PUSH AND PULL MENUS FOR AUDITORY INTERFACES

A Dissertation
Presented to
The Academic Faculty

by

Jeff Wilson

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Computer Science in the
School of Interactive Computing

Georgia Institute of Technology
December 2016

PUSH AND PULL MENUS FOR AUDITORY INTERFACES

Approved By:

Dr. Bruce Walker, Advisor
School of Psychology and
School of Interactive Computing
Georgia Institute of Technology

Dr. Keith Edwards
School of Interactive Computing
Georgia Institute of Technology

Dr. Blair MacIntyre
School of Interactive Computing
Georgia Institute of Technology

Dr. Thad Starner
School of Interactive Computing
Georgia Institute of Technology

Dr. Carryl Baldwin
Department of Psychology
George Mason University

Date Approved: September 26, 2016

To my family
For immeasurable support during my journey

ACKNOWLEDGEMENTS

I would like to express my gratitude to my committee members for taking the time to participate in evaluation of my research. Your efforts are much appreciated and I appreciate your patience and valuable feedback.

Furthermore, I would like to thank current and former student members of the Sonification Lab at Georgia Tech. I have received exceptionally helpful input, advice, and instruction from the team.

Additionally I am very thankful for the support given to me by friends and family. My wife, Becky, has been instrumental in supporting my education goals and has made many sacrifices and taken on burdensome responsibilities to provide me with the time resources to complete this work. My children, Claire and Wyatt, have also been understanding and supportive. I am also grateful to my parents for always supporting my goals.

Lastly, I would like to thank my advisor, Dr. Bruce Walker, for the opportunity to work with him and the Sonification Lab. His dedication to supporting my work has been critical to the progress I have made in pursuing my degree and is very much appreciated.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

TLX	Task Load Index
MRT	Multiple Resource Theory
IVR	Interactive Voice Response
PGS	Personal Guidance System
SWAN	System for Wearable Audio Navigation
DTMF	Dual-tone Multi-frequency
AAC	Augmentative and Alternative Communication
BST	Balanced Binary Search Tree
IVT	In-vehicle Technology
PAL	Platform Abstraction Layer
TTS	Text-to-Speech
API	Application Programming Interface
SAPI	Microsoft Speech API
SOLAFS	Synchronized Overlap-Add Algorithm with Fixed Synthesis
XML	Extensible Markup Language
PCM	Pulse-code Modulated
ANOVA	Analysis of Variance
LCD	Liquid Crystal Display
3VP	Three Vehicle Platoon Task
GSR	Galvanic Skin Response

SUMMARY

Auditory display is an often underutilized interface modality for conveying information to a user. However, audio has previously proven effective in a variety of use cases for information presentation and is particularly effective when the user is unable to attend to a visual interface, whether from a disability or a temporary constraint such as vehicle operation. In addition to auditory representations of data (sonifications), audio can also be used to represent a list of commands or menu within an interface.

This thesis presents a concept for auditory menus that minimizes responses/inputs by the user as well as the number of tactile controls necessary. Such types of menus therefore limit simultaneous manual interactions when the user is also engaged with another demanding motor task. This approach to auditory menu interaction is referred to as a *push menu* and can be thought of as an alternative to more conventional auditory menus, which are referred to as *pull menus*. Push menus present menus in an automated sequence during which the user recognizes the desired menu item and makes a selection within a selection interval. In contrast, pull menus require that the user navigate via a combination of multiple navigation inputs and item selections. In this thesis a general hypothesis is presented that predicts that a primary visual-motor task, such as operating a vehicle, will be less negatively impacted by the secondary task of auditory menu interaction when the menu is a push menu rather than a pull menu.

CHAPTER 1

INTRODUCTION, THESIS STATEMENT, AND CONTRIBUTIONS

Introduction

The auditory modality is often targeted as a means of providing additional information to a user that is engaged in a heavily visual task. For instance, an individual may have difficulty reading while walking given the simultaneous demands on vision, but listening to an audiobook via headphones while walking will likely be found to be a much easier task.

This intuition to leverage sound in interface design when the eyes are busy has been studied in a variety of contexts including studies of piloting aircraft, air traffic control, automobile operation, etc. The use of auditory interfaces has generally been effective but in some cases has been shown to have detrimental effects on the primary task.

Multiple Resource Theory (MRT) (Wickens, 2002) predicts potential benefits when maximizing both visual and auditory perceptual modalities. However, the theory also predicts resource contention at stages of information processing beyond perception. This consideration of multiple stages of processing helps to explain some of the occasionally observed impacts of auditory interfaces on primary tasks.

Auditory interfaces can often be effective for a variety of applications, however the inherently serial nature of audio means that accessing and in particular finding information within a stream of audio a difficult task. This issue is at odds with the fact that an auditory interface that provides a useful set of features needs an efficient means to select and interact with these features. This need leads to the notion of auditory menus and the importance of strategies to overcome limitations imposed by audio’s serial nature.

Perhaps the first auditory menus were realized in the Interactive Voice Response (IVR) industry. These menus are built upon touchtone telephony systems and first introduced a variety of auditory menu concepts that continue to influence auditory interface design in modern devices, such as cell phones and music players. While most individuals are probably familiar with IVRs that have the standard enumerated menu (e.g. "Press one for the first option, press two for the second option…"), other interaction models have also been explored.

One fairly obscure IVR interaction model is unique in that it limits physical interactions of the user. This menu presents options to the user in an automated fashion as is the case with the standard style but there is no enumeration. The user must make her selection during the period of time that the menu item is presented. Without a response, the system continues presenting each menu item one after another. In other words, the user must make her selection within a temporally constrained interval.

While this temporal approach loses some of the flexibility of the standard enumerated IVR menu, it does have a couple interesting characteristics. First, the menu interaction has been simplified to a single button press. A single button is all that is needed to support this interface. Second, the number of manual responses has been drastically reduced as compared to other non-enumerated menu interaction techniques.

Applying MRT to auditory menu interaction for design guidance, one may see potential optimizations of mental processing stages outside of perceptual encoding. In regards to the temporal menu style there are fewer demands on working memory to recall what each button does and which audio events should necessitate which button. Additionally, there are fewer manual responses made as compared to menu interaction models that require button presses to move forward through the menu, a button press to select the current item, and perhaps other button presses to support going backwards through the menu, etc.

In the case of IVRs the benefits of temporal menus generally don't stand out. In fact, the temporal menu style is likely less efficient than the standard enumerated style for IVRs, particularly for expert users. This is especially likely for enumerated menus that allow selection at any time during audio playback of a narration of menu choices. However if a user is engaged in a demanding primary visual-motor task, then the temporal menu type will likely perform well at least in terms of impact on the primary task.

This document proposes a simple paradigm that generalizes menu types originally introduced by IVR interfaces with an emphasis towards modern use cases such as in-vehicle technologies, mobile devices, wearable devices, etc. A distinction is made between menu types that require the user to interactively navigate the menu versus menu types that automate menu presentation such that the user needs only to respond when the desired item is presented. This classification is referred to as the *pull* menu type and the *push* menu type, respectively. This nomenclature suggests heightened demands on the user to request information in the case of pull menus and increased demands on the menu system in the case of push menus.

Within this menu classification scheme, this thesis seeks to address the question of which menu type is most appropriate for situations where a user is simultaneously engaged in a critical and demanding primary task that is visual-motor in nature. Furthermore, can any deficiencies in push menu types be addressed through optimizations of interaction?

Thesis Statement

Leveraging the auditory perceptual modality in interface design has proven to provide an effective means of interaction, particularly when a user has limited or no ability to attend to a visual interface. However, multimodal design decisions that do not fully consider mental resource demands of simultaneous visual and auditory interactions may result in overloading the user. Push menus are an effective technique for allowing users to make interface selections that reduce mental resource contention with other tasks as compared to more typical pull menus. Push menus support menu selection without significant reduction in performance of critical and demanding primary tasks such as vehicle operation.

Contributions

Present an empirically grounded design space of Push Menus

Push menus have been previously discussed by researchers but have largely been within the context of IVRs and with very little consideration of mobile or vehicle applications of push menus. Contributions are made that show a variety of characteristics that can be introduced into menu implementation and adjusted to improve various aspects of performance.

Establish a theoretical basis within the context of Multiple Resource Theory that explains the benefits of push menu interaction during demanding primary tasks.

A theoretical basis is established that predicts the reduced tactile inputs and associated decision making a user makes with push menus relative to pull menus results in reduced negative impact on performance of a demanding primary task.

Develop Flexible and Functional Menu System

A robust audio management platform has been developed which supports complex audio-based applications. A fully configurable menu system has been built as a part of this system. A wearable navigation application has been built that utilizes this system. Additionally, a flexible instrumented user study framework for assessing dual task performance of auditory menus has been built and has supported a number of related studies.

Demonstrate experimentally the effectiveness of push menus in dual task scenarios.

Several user studies were completed that demonstrate the benefits of push menus as compared with pull menus are presented. The first study is detailed with the intent of characterizing push menus by learning the advantages and disadvantages relative to pull menus. The next study places emphasis on analysis of both menu types under demanding dual task scenarios. Two additional studies look at possible approaches to improving push menu performance (accuracy and time to target menu item). A final study was also completed to assess push and pull menus under simulated driving conditions where the user interacts with the menus while operating a virtual automobile.

CHAPTER 2

RELATED WORK

Use of Audio for Conveying Information

The perception of sound offers unique characteristics that can be leveraged to convey information to a user. This psychoacoustic perceptual ability is distinct from the visual modality and presents various advantages and disadvantages in comparison.

While both vision and hearing are critical perceptual abilities, auditory displays are considerably underrepresented in comparison with visual interfaces. Auditory displays can be generally defined as output devices that present audio representations of information to the user. When audio is used in applications outside of multimedia (videos, computer games, etc.), it is often limited to alerts that bring attention to events (e.g. email alerts, input feedback, errors, etc.). One reason for this limited use is that the "transient and temporal nature of audio" (Arons, 1997) makes it difficult to quickly scan to find information. Large amounts of audio can be difficult and time consuming for a user to manage. SpeechSkimmer (Arons, 1997) attempted to address this for speech audio through a variety of processing and interaction techniques but exploring audio with SpeechSkimmer remained a serial task. Text-based visual interfaces supporting equivalent information scanning will often be more efficient when considered as a single primary task.

Beyond the issue of the serial nature of sound, there are also fundamental limitations imposed by "detectability, discriminability, and identifiability" of discrete sounds (Nees & Walker, 2011). These issues are exacerbated by the environment in which audio is generated. Detection of a sound implies that a listener has recognized that a sound has been generated and this ability is limited to a range of sound frequencies and amplitudes. Simultaneous sound sources may mask sounds that would otherwise be heard. Discrimination refers to the ability to distinguish one sound source from another. Sounds that are too similar can be difficult to distinguish. For instance, harmonic frequencies can blend together and be perceived as one sound. Lastly, effective use of audio in auditory displays relies on the communication of information to the user. Therefore, the perceived sound must be identifiable and associated with a learned meaning or mapping.

The "Cocktail Party" (Cherry, 1953) effect describes the ability to listen to several sounds at once and selectively attend to one, ignoring the others. However the cognitive load of simultaneously comprehending two or more speech sources is difficult, given the limited human speech processing capacity (Mowbray, 1953). It is also difficult to use a speech-based interface if for instance a user must carry on a conversation with someone at the same time (Walker & Lindsay, 2005; Wickens, 2000). Therefore, an interface designer must be careful in the way in which speech is used in auditory display. Care must be taken to effectively leverage human speech processing without negatively impacting other interactions outside the interface.

Outside of the use of speech, other types of audio can become attractive for use in interfaces if information can be represented intuitively and concisely. This desire can be addressed by sonification, which encompasses a variety of techniques for creating audio describing sets of information with non-speech audio synthesis. There are many sonification techniques and in particular sonification of graphs is an area of detailed research (Hermann & Ritter, 1999), (Walker & Cothran, 2003). Magnitude of temperature, pressure, size, etc., have been shown to effectively map to sonfications based on frequency, tempo, modulation, and other aspects of audio (Walker, 2007). Related, FiltEars (Cohen & Ludwig, 1991) describes the concept of applying a just noticeable augmentation to a sound that maintains user recognition of the original sound, but that highlights some aspect of the associated information. This approach can be used for describing secondary attributes such as classification group, etc.

An application of sonification has been through efforts to create auditory representations of existing interface components, which are most often represented visually. This can be accomplished through a multimodal and supplementary audio representation, or by fully replacing the visuals with equivalent auditory constructs. There are a range of techniques for supporting interfaces with audio including mapping 2D visual GUIs to audio as with Mercator (Edwards & Mynatt, 1994; Mynatt, 1992) or building auditory menus (Yalla & Walker, 2007). Interfaces for organized lists/menus can also benefit from the use of auditory icons (Gaver, 1986; Marila, 2002) or Earcons (Brewster, Wright, & Edwards, 1993), which are musical patterns of notes that represent an identifier. However, recent work has shown Spearcons (and Spindex) can be more efficient in terms of learnability and recall of mappings (Jeon & Walker, 2009; Palladino & Walker, 2007; Bruce N Walker, Nance, & Lindsay, 2006).

Audio Applications

Audio is well suited to mobile applications due to difficulties of providing a visual display or competing visual demands; therefore disadvantages of audio interfaces become less critical. A variety of research projects have explored audio-based mobile devices. VoiceNotes (Stifelman, Arons, Schmandt, & Hulteen, 1993) investigates a speech and tactile driven, hand-held computer emphasizing easy access to organized audio notes. Nomadic Radio combines synthetic speech, spatialized auditory cues, voice input and audio notification for interacting with information in mobile computing applications (Sawhney & Schmandt, 2000). Nomadic Radio places emphasis on filtering of information sources as well as prioritization of information relative to the user's current task when determining whether to interrupt with notifications or other auditory display.

Some mobile applications have focused on pedestrian navigation and environmental awareness, particularly for the visually impaired. Predominantly, these systems have utilized synthesized speech that speaks instructions to the user. The Personal Guidance System (PGS) (Loomis, Klatzky, & Golledge, 2001; Loomis, Golledge, & Klatzky, 2001) is typical of the modern approaches. The computer creates spatialized audio that seems to come from the same place as the object or feature to which they refer via virtual speech. “Fire hydrant here” would sound as if it came from the real hydrant. Note that PGS has also explored the use of non-speech audio. See, also, the Mobility of Blind and Elderly People Interacting with Computers (MoBIC) system (Strothotte et al., 1996). The Drishti system (Helal, Moore, & Ramachandran, 2001; Ran, Helal, & Moore, 2004) utilizes a voice recognition user interface and synthetic speech output to interface with a geospatial database for navigation aid. Commercial systems developed by Humanware, Sendero, and others work similarly. A System for Wearable Audio Navigation (SWAN) (Wilson, Walker, Lindsay, Cambias, & Dellaert, 2007) however places emphasis on non-speech sounds, including auditory icons (Gaver, 1986) and spearcons (Bruce N Walker et al., 2006). In the context of navigation cues, studies have shown that using non-speech audio leads to better performance than speech cues in situations where there is a significant cognitive load from performing another simultaneous task (Klatzky, Marston, Giudice, Golledge, & Loomis, 2006).

Conveying information that is positional/geospatial is often critical for mobile applications. Spatialization of audio describes the process of simulating sounds as if they are emanating from 3D positions. Generalized head-related transfer functions (HRTFs) (Wenzel, Arruda, Kistler, & Wightman, 1993) can be used for generating spatialized audio. Spatialization is useful for representing data that has an associated position (e.g. geospatial data), or for improving the ability of listeners to isolate the source (discriminability). Spatialized audio can also contribute to the immersion of a user in a virtual environment (MacIntyre & Feiner, 1996). Note that continuous tracking of the user’s head position and orientation are necessary if the user is mobile, though the requirement can perhaps be relaxed to only tracking position and orientation of the torso for certain use cases.

While there is a lot of interesting work in presenting information with audio and developing applications that are predominantly audio-based, including mobile, most work so far has looked at standalone applications or capabilities that are accessed one at a time and without much concern for the impact on other tasks a user may be simultaneously performing.

Auditory Menus

While not as pervasive as visual menus, auditory menus are fairly common in a variety of audio-centric applications. Perhaps the most well known example of an auditory menu can be seen in interactive voice response (IVR) systems for use with touchtone telephones. IVRs allow a user to receive automated prerecorded and/or generated audio information (predominantly speech-based). Additionally the user can send information by way of touch-tone button presses. Later IVRs have also supported voice input. Shortly after the advent of dual-tone multi-frequency (DTMF) signaling in telephony systems of the early 1960's, the first IVRs were introduced in the banking industry to support checking balances on financial accounts (Dowburd, 1994). The original IVRs were extremely limited in capability. However, in the 1980's IVRs became much more capable due to advances in hard drive storage allowing more digital audio to be stored and accessed. This improvement in technical capabilities led to progressively more complicated IVRs necessitating the need for improved approaches to menu interaction. IVR applications include phone banking, directory services, voice mail, teletex (text, document access), etc.

Initial IVR development was largely a commercial endeavor, but IVRs have been heavily studied including efforts to characterize the design space of IVR interfaces encompassing presentation of lists, menus, and interactive forms (Resnick & Virzi, 1995). IVR menus are often implemented with absolute mappings to touch pad labels “1” through “9” (“0” is often reserved to request the operator, and “*” and “#” reserved for special commands). The menu is presented as a continuous narration of menu choices with corresponding numerical labels (e.g. “To choose option one, press one. To choose option two, press two…”). This narration can generally be interrupted at any time with a user input selecting the desired option, but sometimes a user is not allowed to make a selection until the end. Resnick refers to this menu interaction type as “standard style”, but the technique has also been referred to as “enumerated” (Schmandt, 1993).

A second menu interaction technique is also described that is less common than standard, but is used in some more complex IVRs such as advanced voice mail systems. This style is denoted as “two-button style.” The two buttons are in reference to one button to select the current menu item and a second button to advance the current menu item to the next. Both selection and advance inputs can interrupt audio playback so that the IVR can immediately respond. The two buttons can be expanded upon by adding an additional button to navigate in reverse as well as another to back out of hierarchical sub-menus. While these new buttons obviously result in an increase of total buttons, users have been observed to predominantly use the two primary buttons (Resnick & Virzi, 1992), hence the name.

Another menu option is possible in IVRs called a “temporal menu” (Schmandt, 1993). A temporal menu is a one-button menu which presents menu items automatically in sequence and the user must make a selection during playback of the desired item. If no selection is made, the menu continues presenting each item automatically. This is similar to the standard/enumerated style discussed previously but without the absolute mapping to multiple numerical inputs that can be chosen at any time (e.g. selection of an item can only occur when that item is active). Resnick generalizes these menu types to define two dimensions describing IVR menus. The first dimension is selection technique: absolute numeric or positional. The absolute numeric menu type describes the standard/enumerated type of menu. Positional menus are menus where the user's inputs for selection are dependent on the current playback position. The second dimension is the method of menu advance: timeouts, skip key, or timeouts and skip key. This dimension details what event(s) causes the transition from one menu item to the next. A timeout without input triggering advance is one possibility, or an explicit input event is needed to advance. Lastly, both event types can be supported. Figure 2.1 from Resnick (1995) shows the possible interactions from the design space defined by the two dimensions.

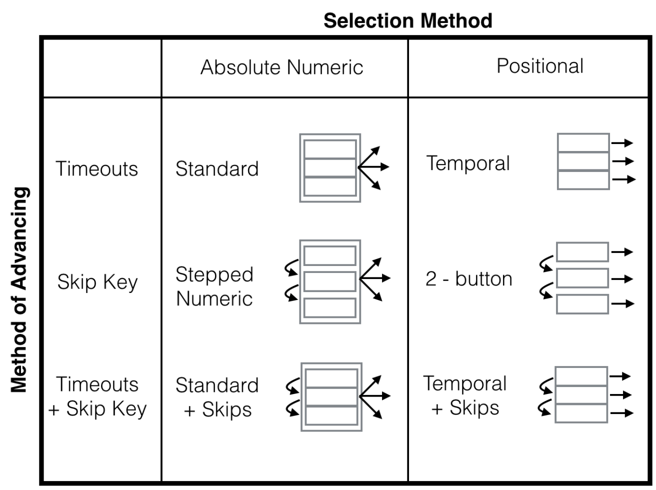

Figure 2.1 Resnick (1995) IVR menu interaction possibilities.

Outside of IVRs, auditory menu design has largely focused on the two-button style. Absolute numeric selection is likely a consequence of the telephone button form factor and also constrains the number of menu items a flat menu can contain to the number of buttons, unless multiple digit numbers (and therefore additional button presses) are supported.

Omitting absolute numeric selection (a special case primarily suitable for IVRs), only the method of advancement remains as a dimension of the design space of auditory menus. These advancement methods include: temporal, two-button, and temporal with skips. The three methods can be reconsidered in terms of the way in which information is presented to the user. In the temporal case, the presentation of information is automatically *pushed* to the user. In the two-button case, the user *pulls* the information from the menu. These two paradigms can be described as *pull* and *push*. The temporal with skips style is a hybrid of the two.

There are relatively few examples of implemented push menus. One notable example is the scan mode found on many car stereos. Typical car stereos have a “TUNE” control knob for fine tuning radio frequency, a “SEEK” button that automatically jumps to next station (next tunable frequency) then stops, and a “SCAN” button. When pressed, the “SCAN” button jumps to next station, offers a brief audio preview, and then continues this process until the user again hits “SCAN” to make the final selection. This scan feature demonstrates the pushing of information to the user and which requires that the user make a selection within a selection interval. Apple iPod Shuffles also demonstrate a very simple push menu. Fourth Generation iPod Shuffles have a small button that when tapped will recite the name of the currently playing musician and song. If this button is pressed and held, then a push menu will begin listing all playlists that have been synced to the device. A user can press the button a second time to select the desired playlist when it is recited.

Curo Interactive is noted for development of a prototype one-button cell phone that relies on push menus, but has not penetrated into the commercial market. Finally, push menu concepts have been applied to assistive technologies such as augmentative and alternative communication (AAC) technologies (Colven & Judge, 1996). For reference, such interfaces have been popularly demonstrated by Stephen Hawking and aid him in communicating. Words+ E Z Keys is a configurable interface for assisting users with text typing on personal computers. Some users with significant mobility constraints may be restricted to single binary physical input interactions (e.g. eyebrow switch). In these cases, push menus are appropriate. E Z Keys works by representing keyboard keys in a hierarchical push menu. Users must make multiple keypress inputs to narrow the hierarchy down to a single selection. When final selection is made, a virtual keyboard key press is generated.

Multi-Level Auditory Menus

Given that in general a menu can be of arbitrary length, and possibly quite long, care must be taken with a push (or pull) menu so that a user does not have to wait an unnecessarily long time to make a selection of the desired item. One may argue that a menu should be a carefully designed static hierarchy of more manageable submenus, should the need arise. A static hierarchy provides the user with an interface that is learnable and menu item locations are memorized. However, in some cases menus may need to be dynamic and not lend themselves to well-defined logical grouping. For instance, a list of contacts in a cell phone can quickly become a challenge to navigate as it grows. Alphabetical ordering can help but the most commonly selected contacts can be shuffled to the beginning of the list, which may be more useful to the user. In some cases, a user may make the most efficient selection of a menu item by executing a multitude of selections in a divide and conquer approach where the hierarchy is itself dynamic as items are added or removed from the menu.

Consider a menu of N items that can be logically ordered (in alphabetical order, for instance). If one assumes that a user has equally likely probabilities of selecting any one of the menu items at any given time, then a very time efficient selection method is that of a balanced binary search tree (BST). Multiple menu selections must be made at each node until the desired node is reached. The choice at each node in a BST typically will be ternary, but can also easily be binary (an approach often used by AAC devices). However, previous work (Miller, 1981) has demonstrated tradeoffs of breadth versus depth for menus in general and indicates that depth causes more user error than breadth. In particular, (Boren, Moor, & Anderson-Rowland, 1997) found binary menus to be less effective than four or eight choices per level in telephone menus. Therefore, it may be necessary to use a balanced tree structure of a different number of branches for optimality.

Unfortunately, in some cases an auditory menu may not lend itself to logical ordering that facilitates use of tree structures. Alternatively, requiring that the user determine menu item membership in the subgroups may be prohibitive. Can a user easily determine whether a word fits into any logical sub-grouping based on alphabetical order or other scheme?

Context is also a powerful tool for optimizing menus. In many cases, context may provide cues that allow the menu system to prune or reorder menu presentation in efficient layouts. For instance, menus may calculate which items are most frequently used and reorder menus so that those items are presented first. Another example is auto-completion when menus are used to select letters when spelling words. This same spelling task menu system can also reorder alphabet letter presentation based on the likelihood of any given letter following the previously entered letter using a dictionary for context. Additionally, menus can provide context to the user to make her navigation more efficient. Earcons (Brewster et al., 1993) and Spearcons (Walker et al., 2006) are both techniques with auditory menus that can be used for this purpose by providing cues and are perhaps most beneficial for pull menus coupled with preemptive inputs.

Use of Auditory Displays in Vehicles

In recent years, in-vehicle technologies (IVTs) have become pervasive as vehicle manufacturers have transitioned from only providing transportation to also providing a software, hardware, and service platforms to provide a variety of functions such as geographic navigation aids, accident avoidance functionality, infotainment, etc. (Nees & Walker, 2011). There are serious safety implications for these sources of distraction for automobile operators. While detailed guidelines are in place for visual interfaces, there is very little guidance for auditory displays(National Highway Traffic Safety

Administration, 2012a). For instance, there are specific guidelines for the placement of LCD panels in car dashboards relative to the windshield and height of the car (National Highway Traffic Safety Administration, 2012b). Additionally recommendations are made that tasks should not require glances away from the road of more than 2 seconds and no more than 12 seconds cumulatively. However, there are no similar guidelines for auditory interfaces. Researchers have begun to identify some guidelines such as for collision avoidance alarms (Baldwin, 2011).

There are well-understood challenges facing the use of audio in IVTs. The previously mentioned challenges of detectability, discriminability, and identifiability of discrete sounds (Nees & Walker, 2011) are exacerbated by the environment of an operated vehicle. Sounds that may compete with the auditory displays of IVTs include wind and road noise, alerts/alarms from disparate vehicle systems, music, conversations, etc.

Designers of IVTs utilizing auditory display should ideally take a holistic approach to interface design and consider that all vehicle systems should coordinate auditory display. For instance, an IVT may prioritize a low oil alarm or collision avoidance alert over navigation instructions. Music may be paused or muted to alert the driver to upcoming road debris. In some cases, low priority audio may be canceled. Other times deferment to the next available time would be more appropriate. Additionally, vehicle speed and throttle position may be used to dynamically set auditory display volume at an intensity that supports detection, discrimination, and identification.

Personal communication and computing devices are also commonly used by drivers while operating vehicles. These devices may offer some features that benefit drivers' ability to maintain focus on driving. For instance cell phones may offer hands free capabilities. Although studies have shown that even hand free calls negatively impact driving (Horberry, Anderson, Regan, Triggs, & Brown, 2006). Some devices may benefit from contextual interfaces that detect that an individual is likely driving and adapt to the driver's diminished visual-motor ability to attend to the device. For instance it has been suggested that a wearable computer could detect vehicle operation and switch to an audio-only interface (Starner, 2001).

Multiple Resource Theory

Multiple Resource Theory (MRT) is a theory of human information processing and predictive framework for making relative comparisons of expected task performance when multiple tasks are performed concurrently (Wickens, 2002). MRT expands upon time-sharing models (Kahneman, 1973) that only consider overall time costs, but not the mental resources now believed to be largely independent. MRT breaks cognitive resources into three stages of processing (perceptual encoding, central processing, and response). Across these stages are the visual and auditory modalities. Within these modalities information can be coded spatially or verbally. For instance, a siren may be perceived with the auditory perception as a non-verbal sound with spatial location information. However, a speech warning via loudspeaker would be processed verbally. A flashing light would be perceived spatially within the visual modality, whereas text on a screen is considered visual-verbal. Visual encoding has also been shown to further divide into parallel processing of foveal and peripheral information.

In practice, MRT is best suited for predicting resource contention at the perceptual encoding or response stages. Except for very constrained tasks such as mental rotation of visual figures, it is very difficult to predict which mental strategies are in use by individuals and can therefore vary greatly.

Manual response resource contention is of particular interest. Individuals are often observed conducting two or more demanding physical activities that may appear to be executed in parallel. For instance, a drummer may maintain different beats with the left and right arms as well as the feet. Previous research suggests that such complicated simultaneous manual tasks can only be completed through an integrated sensory-motor representation of the individual manual tasks (Klapp & Nelson, 1998). In the case of the drummer, the individual rhythms are incorporated into one pattern. Therefore concurrent manual response demands are likely fulfilled by potentially costly context switches back and forth between the contending tasks.

The auditory perceptual modality has been observed in some dual task scenarios to have a preemptive effect on simultaneous tasks (Wickens, Dixon, & Seppelt, 2005). In response to an audio event, a user may fully focus on the audio to the detriment of other tasks. This preemptive effect may generally be more likely for an alert rather than audio presented at the user's prompting (e.g. the user explicitly engages an auditory menu whereas an alarm is unexpected).

CHAPTER 3

THE SWAN MENU SYSTEM

The auditory menu features for push and pull menus are implemented on a custom application originally developed for the System for Wearable Audio Navigation (SWAN) (Wilson et al., 2007).

The System for Wearable Audio Navigation (SWAN) is a navigation aid for individuals who have suffered physical vision loss (e.g. full or partial blindness) as well as those affected by temporary loss of vision such as firefighters in a smoke-filled environment.

SWAN addresses the limitations of previous speech-based navigation aids by using non-speech audio presentation of navigation information whenever possible. SWAN provides an auditory display that enhances the user's ability to (1) keep track of her current location and heading as she moves about, (2) find her way around and through a variety of environments, (3) successfully find and follow a near optimal and safe walking path to her destination, and (4) be aware of salient features of her environment.

SWAN supports these goals through sophisticated position tracking technologies, sonification of navigation routes and environmental features, and implementation of a database of information relevant to the user's navigation needs. SWAN allows users to record their movements or paths through the environment. These paths are used to create a personally relevant set of maps for the user. Additionally, the user can annotate objects found within the environment including locations, features, and obstacles. This could include, for example, a particular bus stop, a favorite coffee shop, or a section of sidewalk prone to flooding after rain showers. The map can also be queried for directions to a particular location. To indicate paths and features in the environment the user is presented with a set of non-speech audio cues that guide them along the paths, all the while sonifying features and obstacles the user encounters.

The goal of SWAN's menu system is to provide context aware interaction schemes that are optimized for the workload and mobility of the user. From this initial effort, push menus were explored as a promising interaction technique for high workload demands, particularly in dual task scenarios. Given encouraging informal results, a course of study was pursued as detailed in this document.

The experimental system for push and pull auditory menus is built upon core audio components of SWAN. Those components are detailed below.

SWAN Audio Components

SWAN features several layers of components in support of robust audio presentation. The first is a platform abstraction layer (PAL). The PAL provides a standard interface to SWAN for access to hardware audio buffers as well as buffer attributes such as 3D spatialization, attenuation, frequency control, etc. Currently the PAL is build on top of OpenAL, which is itself a standardized interface, but other PAL interfaces such as DirectSound could be easily supported.

Leveraging PAL, a higher-level AudioEngine is implemented. The AudioEngine's primary task is the management of system resources relative to demands. For instance, it is not uncommon for an application to send requests to render many simultaneous sounds. These demands can easily surpass system capabilities, particularly on mobile platforms. Therefore, it is important for the AudioEngine to make decisions on which audio playback requests are serviced. Typically, audio APIs either return an error when too many simultaneous audio buffers are requested and/or only support a first come, first serve buffer allocation policy. SWAN's AudioEngine supports a more advanced buffer management scheme.

Given that spatialized audio is a critical component of SWAN, the AudioEngine is able to assess distance-based attenuation parameters in real-time. If a sound source is too far to be heard from the listener position, the buffer is automatically reclaimed and can be allocated to new sound source requests. A hysteresis feature is implemented to limit situations where a source oscillates between the edge of receiving allocation or having its existing buffer revoked. If too many sources are within range, only the nearest sources are provided with playback buffers.

To better support spatialized audio buffer management a new attenuation model was developed to avoid harsh cut-offs when a source is far enough away for the AudioEngine to revoke the buffer. The AudioEngine builds upon the *inverse-distance* attenuation model used by most audio APIs and hardware by adding an additional linear fade-out once the gain factor drops below a certain threshold. This fade-out is a function of distance and brings the sound source to zero gain (completely inaudible) before the hardware buffer is revoked. This approach avoids pops and unpleasant dropouts of audio. Similarly, a fade-in occurs when a buffer is granted to a source.

In addition to the AudioEngine's buffer management for spatialized audio, it also supports a priority system that applies to both 3D and non-3D sound sources. The priority system inspects a numerical attribute assigned to each sound source that specifies a priority level. A lower number denotes a higher priority. The AudioEngine services all sound source requests at the highest priority, then goes to the next priority, etc.

This simple priority mechanism allows the SWAN application the ability to service important events, such as alarms or alerts, without fear that those critical audio clips go unplayed due to other less important sounds taking up all the hardware buffers. Furthermore, the on-demand buffer management of the AudioEngine is much more effective than for instance assigning one buffer to alarm sounds and nothing else. As alarm playback is typically a rare occurrence, a dedicated buffer would make inefficient use of audio buffer resources.

The AudioEngine also supports a Text-To-Speech (TTS) source, which streams text to an underlying TTS rendering engine that then passed rendered speech audio to an output buffer. The TTS source supports attributes such as speed (e.g. words per minute) and hints for voice selection (e.g. male, female, dialect, etc.). The current TTS source implementation interfaces with Microsoft Speech API (SAPI) and Cepstral voices (https://www.cepstral.com/).

Another custom audio source is implemented to support a period of silence. This source is useful for creating compound sounds in the auditory menu system (see below) with appropriate spacing.

SWAN leverages the AudioEngine to provide application features such as navigation aids, environment descriptions, and auditory menus (to access various application features). Each of these features utilize the AudioEngine's capabilities such as advances buffer management to provide system functionality with efficient use of audio hardware and graceful degradation when those resources become limited.

Auditory Menu System Technical Design

SWAN's auditory menu system is built around a custom state machine and event queues for playback and input handling. The menu system can be reconfigured on the fly to function as either a push or pull menu (or a hybrid of the two). The state machine processes playback events stored in a play-queue. These events are handled one at a time and the state machine plays each requested playback event in order once the current play request is complete. The play-queue allows compound sounds such as intro/vocalized menu item/outro. Play events can be marked as interruptible or not. This functionality can be used to allow a pull menu that can be interrupted to skip to the next menu item before playback of the current item is complete. Additionally, some play events such as a special sound that denotes that the end of the list has been reached cannot be preempted.

The menu system also processes an input-queue that is checked regularly for any new input events such as menu item selection, moving up or down within the menu, etc. These input events can generate immediate audio feedback to acknowledge user input. Additionally, the inputs can cause interruption of the play-queue, causing all queued interruptible play events to be purged (such as the case of skipping to the next menu item). The play-queue is then filled with play events that represent the next menu state.

Supplementary Audio Tools

Additional tools were developed to support advanced features such as applying pitch corrected time compression to audio used in menus as well as removing leading and trailing silence from clips and replacing with silence of a known length. A low level C++ library was developed that includes implementation of a streaming version of the Synchronized Overlap-Add Algorithm with Fixed Synthesis rate (SOLAFS) (Henja & Musicus, 1991) as well as basic Wave file editing and manipulation. A simple command line application was built that used the library to generate menu items of various playback rates. Additional features allow the generation of spearcons (Walker et al., 2006) from recorded speech (for synthesized speech rate adjustment is best applied within the speech synthesizing software). These spearcons can be added to pull menus within the SWAN menu system. Alternative spearcons with variable time compression rates favoring less compression at the beginning and more towards the end are possible with both linear and logarithmic models. The C++ library is designed to operate on real-time streams with low latency but has not yet been integrated into the SWAN Audio Engine.

Study Software

The SWAN auditory menu system is able to effectively support menu requirements for the features of SWAN. However, considerable work was necessary to adapt the menu system to support the studies of push menus.

First a visual component is implemented that provides a synchronized visual analog to the auditory menus. This visual menu interface is implemented in Microsoft .NET and shows paginated menu items along with highlighting of the current menu item. The visual display can be adapted for a variety of monitors, including vehicle displays.

An XML (Extensible Markup Language) format has been developed that allows complete experiment configurations to be developed without modifying code or rebuilding the experiment software. This configuration supports building studies with multimedia instructional pages, input hardware selection, selection of experiment conditions, assignment of experiment blocks and trials, randomization of blocks/trials, etc.

The randomization capabilities allow automatic randomization of experiment blocks (to counter order effects), randomization of menu order, and random selection of trials. The same randomized menu can optionally be frozen across multiple blocks and these blocks can also be optionally grouped together. Trial randomization is implemented via a binning procedure, which enforces a semi-uniform sampling of target menu items across the entirety of the menu. For instance, a menu of forty items with a block of ten targets will result in the forty items being dividing into ten bins of four items each. Up to four blocks can be randomly generated from these bins randomly removing one item from each bin.

Additionally, a full version of NASA TLX (rating scale rankings and estimates) is implemented within the experiment software and is built using the SWAN auditory menu system and a custom graphical view. A physical copy of the NASA TLX rating scale definitions is intended to be provided to participants along side the software questionnaire.

Primary Task Game

Some studies of push/pull auditory menus utilize a primary task meant to approximate the task of operating a vehicle. This primary task is implemented as a simple computer game in which the participant must catch balls falling from the top of the screen.

The ball drop game was developed in the Terathon C4 Game Engine. The game appears on screen as X columns in which balls fall from the top of the screen at a constant speed. The player must move a ball catcher (or paddle) from column to column to catch the balls by pressing the left and right arrows on a computer keyboard. Note that the balls do not accelerate as they would under simulated gravity. Instead they move in a downward direction with constant velocity.

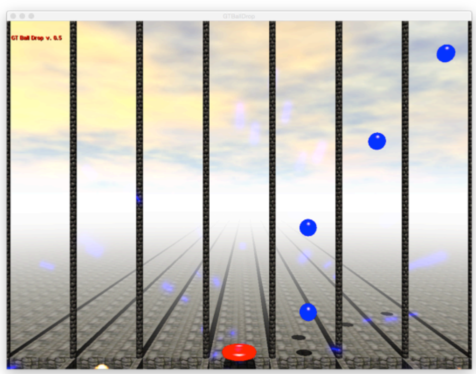

Figure 3.1: Screen capture of the ball drop game.

The game is configured so that the balls that fall are always no more than one column away from the previous ball to fall, but otherwise randomly determined. In other words, if a ball is falling from a given column, the next ball can come from the same column, one to the left, or one to the right (with configurable likelihoods). If the current column is to the far left or right, then the next ball can only fall from the same column or the only adjacent column. This pattern was chosen as it roughly follows a lane holding task when operating an automobile. (An optional mode to drop a ball from a random column also exists.) Players receive visual confirmation of a successfully caught ball or a missed ball. A successful catch is rewarded with a small fireworks explosion at the point of impact. A missed ball appears to burn up at the bottom of the screen.

The ball drop game has an adaptive difficulty calibration feature, which is used to normalize difficulty across participants. This calibration is performed by itself, and not under dual task conditions. The calibration step works by automatically adjusting ball speed and time between ball drops until the target mean accuracy is achieved over blocks of a certain number of trials (ball drops). The adaptive adjustments begin with an initial step size for both parameters. The initial state is designed to be extremely easy. A block of trials is presented to the user. If the user has an average success rate above the target success range (between minimum and maximum allowed success rate) then the parameter difficulty steps up. However, if the user's average is below the target range then the step size is halved and the difficulty is reduced by changing the sign of the step and adding to the existing parameters. If the user's average is within the target range then the correct parameters have been determined. Every time the direction of difficulty stepping reverses (sign changes) from the previous block, the step size is halved. A maximum number of step reversals are enforced; even if the user's average success rate is not within the target range, the final parameters will be locked in. During calibration minimum time between ball drops and ball speed are capped to avoid degenerate cases (such as reversed velocity direction or overlapping balls). A typical target success rate for dual task studies is 85% (+/- 5%).

Once these game parameters are determined for a participant, they are preserved and used for the actual studies. The reason for this normalization of difficulty is to ideally avoid ceiling effect situations were participants have extra capacity to maintain peak performance in the game even when their attention is divided in the dual task study. The assumption is that if the game is not sufficiently difficult, potentially no observable drop in game performance will ever be observed as different auditory menu conditions are delivered. Also, the normalization avoids selecting a single difficulty setting for all participants, which may be overwhelming for some and too easy for others.

The ball drop game supports a remote trigger option via a network socket interface. This allows precise timing data in both the ball drop game as well as the auditory menu experiment software.

Log Analysis

Custom log processing and analysis scripts were written in Octave, which is an Open Source equivalent to MatLab with very similar syntax. The Octave scripts allow for precise segmentation of log events. For instance, in dual task studies primary task log events can be segmented into on-trial or off-trial to allow analysis of fine-grained aspects of primary task performance.

Push and Pull Auditory Menu Features

As mentioned previously, the SWAN Menu System supports a real-time switch between push and pull interaction types. This can be triggered by a context-aware event (e.g. noticing that the user is engaged in a primary task such as walking) or explicitly, as is the case with the user experiment software layer.

The menus are highly configurable and there are many options that affect presentation and interaction with the menus. For instance, intro/outro's, separators, etc., can all be added to menu items. This enables features such as audio icons, Earcons, Spearcons, etc. Additionally, rollover from the end of a menu list back to the beginning (along with playback of audio denoting rollover) or vice versa can be enabled or disabled.

Push menus possess a number of configurable options. The first option is the silence interval. This silence interval specifies the silent period between menu items during which an item can still be selected before the next item plays automatically.

Another option is the selection overlap. This is a typically short period of time during active playback of an item that a user selection will actually result in the previous item being selected. The selection overlap is implemented with the hypothesis that a certain amount of time is necessary to process and recognize an item. Therefore, a selection event that occurs during this time is likely to be intended for the previous item. Special cases are implemented for the first item (no overlap rule imposed) and last item (extended silence interval equal to normal silence interval plus selection overlap).

Push menus can also be configured with varying interactions. These include a button press that allows the user to back up one menu item if she overshoots the target. Also available is a Push/Pull hybrid interaction scheme that allows the user to choose the direction that the menu is presented to the user. This is somewhat akin to a fast-forward and rewind button on a digital video recorder. Push menus can also be configured for the number of times looping through the menu before timing out.

CHAPTER 4

USER STUDY 1 - INITIAL CHARACTERIZATION OF MENU TYPES

The first experiment sought to characterize fundamental differences between pull and push menus. Participants were given a target menu item which they were tasked with finding in either a pull or push menu. A visual menu that coincided with the auditory version was presented to half of the participants as a between subjects factor.

Several dependent variables were measured including menu selection accuracy, trial duration (time until selection made), workload, preference, etc. Related to these variables, there were several hypothesized outcomes. First, menu selection accuracy was expected to be worse, trial duration longer, and workload higher, for the push menu type. Also, accuracy was expected to be worse, duration longer, and workload higher, as menu size increased for either menu type.

The visuals-off selection accuracy was expected to be worse, trial duration longer, and workload worse than visuals-on.

Study 1 - Participants

Twenty-nine undergraduate students (after one exclusion: 15 male, 13 female; mean age = 20, SD=1.5) participated in a study for credit in a psychology course at the Georgia Institute of Technology. Participants gave consent to participate in the study. All reported normal or corrected-to-normal vision and hearing. One participant was excluded due to software error.

Study 1 - Apparatus

A Dell personal computer was used with monitor Dell CRT 17-inch monitor running at 1024x768 resolution. Sennheiser HD202 Headphones were worn by the participant. A Microsoft Xbox360 wired USB controller was used for participant input. The experiment was conducted in a quiet office.

The menu software was written in Microsoft Visual Studio .NET C# using a modified version of the audio engine and auditory menu system from the SWAN project (Wilson et al., 2007). A one-dimensional vertically oriented graphical menu was also developed to support the visual condition. Please see previous discussion of the SWAN menu system for further details.

Both the push and pull menus use the same content, which are lists of similar items. All lists were generated using Cepstral Text-to-Speech ("Cepstral - Text to Speech for Personal, Business, and Enterprise Use," n.d.) synthetic voice, "David" (male, USA English), with automation driven by Microsoft's .NET Speech API (SAPI) generating spoken word from text strings and saving to pulse-code modulated (PCM) Waveform Audio File Format (WAV) at 44.1 kHz sampling rate and 16 bits per sample (monaural). In cases where Cepstral provided incorrect pronunciation, the associated text strings were modified to provide a phonetic representation of the desired output in an iterative trial-and-error approach until the desired output was achieved. It was found that the generated audio was framed by an inconsistent amount of leading or trailing silence. A dedicated program was developed that characterized the leading/trailing silence and normalized the lengths of silence across all the generated items to 15 milliseconds. Additionally, all files had volume levels normalized using the audio editing program, Audacity ("Audacity: Free Audio Editor and Recorder").

A list of forty unfamiliar names was generated via an online random name generator as described in (Palladino, 2008). A list of twenty animal names considered familiar was generated as well. Lastly two lists of length five were generated. The first was a list of writing tools. The second was a list of everyday objects. These lists served as the content for menus of the same length.

Study 1 – Design and Procedure

Within-subjects conditions are the three menu sizes (40, 20, 5) crossed with the two menu interaction types (pull and push) and paired blocks. The visual menu condition was assigned as a between-subjects condition. Participants were randomly assigned to all push experiment blocks first, then all the pull experiment blocks, or the opposite. Two versions of the experiment software were configured to support this randomization, and assigned by coin flip. The visual condition was configured and assigned in a similar fashion.

After obtaining informed consent, the experiment administrator guided the participant to experiment area. The participant was shown how to adjust and wear the headphones. A test sound was played for the user and if necessary the volume was adjusted to a comfortable level. The participant was also told to use the Xbox 360 controller and view the computer screen for interactive instructions and experiment. The participant was left alone to complete the computer portion of the study, which was fully automated. Upon completion of the study, the participants filled out a paper questionnaire regarding demographics and preferences.

The software consists of an introduction to the study, training in the use of pull and push menus, an introduction to NASA TLX workload assessments, and the actual experiment. The participant was able to use the controller to move through the introduction as a series of screens with static text and image content. Each screen has options of going forward or back or reading aloud of the onscreen text. In some cases a participant is not be able to go back through the screens, such as after going through a training session or the actual experiment blocks.

The training consists of menus identical in form to the actual experiment but with unique content. In this case, the content is a twelve-item list of people, places, and things well known on the Georgia Tech campus. Participants were always trained with pull menus first then push menus on the assumption that it would be best to first introduce the interaction that the participant would likely be more familiar with before the less familiar interaction. Training always matched the visual condition assigned to the participant.

Whether assigned a visuals-on condition or visuals-off, the top of the screen always shows "Target:" to continuously indicate the current target. When a new menu target is presented, the participant hears, "Target is <target_word>." Visually the user sees "Target: ???" as the phrase "Target is…" is narrated, but before the actual *<target_word>* is spoken. Once the *<target_word>* is stated the "???" on the screen is replaced with the text of the word. This is done to make sure that participants cannot read ahead of the narration presenting the target. During this sequence of presenting the target, menu interaction is locked out for the participant. (Attempts to interact with the menu while locked out results in a buzzing sound.) When the "Target is <target_word>" completes playback the background color behind the line of text changes color to indicate that the participant can now begin navigating the menu. For either visual condition, the target remains on the screen for the duration of the trial in case the participant forgets what the target is.

For the visuals off condition, the rest of the screen remains blank. For the visuals on condition, the screen shows up to eleven menu items with the currently selected menu item highlighted. For menus of a greater length, "more" plus a down arrow appears in the bottom right of the screen. If a participant navigates past the last item on the screen, the entire screen refreshes to the next section of the menu and the highlighted item will now be the first at the top of the screen. If the menu is showing the second or greater screen of the menu, then "more" with an up arrow appears in the top right of the screen.

Each within-subjects condition corresponds with two consecutive experimental blocks of ten trials each. The menus are pseudo-randomly ordered by the experiment software for the pair of blocks (same menu order for both blocks). Ten menu targets are selected randomly from ten equal-sized, ordered bins of the menus for the first block, and then ten more selected of what remains in the bins for the second block. In the case of the menu of size five, four consecutive blocks are presented to a participant using two different size five menus. Each of these blocks has five trials. These smaller blocks are paired to create a larger block with the same number of trials as blocks for the larger menu sizes. This pairing is done in order to more easily compare against the other menu sizes and still control experiment conditions.

Before each block, a menu configuration overview screen is presented detailing menu interaction (including reminder of buttons to use), menu size, number of targets, and any menu-specific parameters, such as the silence interval between each item in the push interaction condition.

After each pair of blocks (or four small blocks for the menu size five conditions), a NASA TLX workload assessment is conducted within the experiment software. NASA TLX dimension weightings are assessed by the user for every workload assessment.

Study 1 - Results

Results were analyzed with a 2 (push or pull menu type) x 3 (menu size: 5, 20, 40) x 2 (block: A,B – ordered blocks of same conditions) repeated-measures analysis of variance (ANOVA).

Statistically significant differences were found between menu types in mean accuracy of selection, $F(1,26) = 18.64$, $p < .05$, $\eta_p^2 = .42$ mean duration of trial, $F(1,26) = 1154.59$, $p < .05$, $\eta_p^2 = .98$, and mean perceived workload, $F(1,26) = 7.43$, $p < .05$, $\eta_p^2 = .22$. See Table 4.1 for descriptive statistics.

Table 4.1 Descriptive Statistics

Menu Type	Accuracy		Trial Duration (sec)		Workload	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Pull	.99	.02	6.98	1.35	17.77	10.68
Push	.97	.03	16.39	1.25	23.14	13.42

Also, statistically significant differences were found between menu sizes for mean workload $F(2,52) = 25.77$, $p < .05$, $\eta_p^2 = .50$. Trial duration significant differences were present as well, $F(1.23,32.06) = 2071.99$, $p < .05$, $\eta_p^2 = .99$, corrected with Huynh-Feldt for sphericity violations. Mean accuracy differences for menu size were not significant, $F(2,52) = 2.09$, $p = .135$, $\eta_p^2 = .074$.

Paired-sample t-tests were conducted, with Bonferroni adjustment for Type I error, for the statistically significant main effect of menu size on trial duration and workload. Resultant p values from t-tests are multiplied by the number of comparisons (3) and maintaining the same critical alpha of .05, which is the equivalent alternative to correcting the critical alpha to .05/3 = .017. Refer to the method used in SPSS ("IBM The calculation of Bonferroni-adjusted p-values - United States," 2012). Menu Size 5 trial duration ($M = 2.99$, $SD = 0.3$) was smaller than Menu Size 20 trial duration ($M = 8.73$, $SD = 0.94$), $t(28) = -36.59$, $p < .05$, as well as Menu Size 40 trial duration ($M = 23.34$, $SD = 2.34$), $t(28) = -47.99$, $p < .05$. Menu Size 20 trial duration was smaller than the Menu Size 40 trial duration as well, $t(28) = -43.20$, $p < .05$. Menu Size 5 workload ($M = 14.16$, $SD = 11.00$) was smaller than Menu Size 20 workload ($M = 19.54$, $SD = 11.60$), $t(28) = -4.23$, $p < .05$, as well as Menu Size 40 workload ($M = 27.66$, $SD = 14.30$), $t(28) = -6.08$, $p < .05$. Also, Menu Size 20 workload was smaller than Menu Size 40 workload, $t(28) = -3.96$, $p < .05$.

There was a statistically significant interaction between the menu type and menu size on mean trial duration, $F(1.41, 36.58) = 651.35$, $p < .05$, $\eta_p^2 = .96$, corrected with Huynh-Feldt for sphericity violations. Trial durations for menu type and menu size are shown in Table 4.2 and Figure 4.1. Refer to Appendix B for details of pair-wise Bonferroni-adjusted t-tests, which were all statistically significant.

Table 4.2 – Menu Type * Menu Size Descriptive Statistics

Menu Type	Menu Size	Mean	Std. Dev.
pull	size 5	2.23 s	0.42
	size 20	5.36 s	1.40
	size 40	13.35 s	2.80
push	size 5	3.75 s	0.38
	size 20	12.11 s	1.06
	size 40	33.32 s	2.96

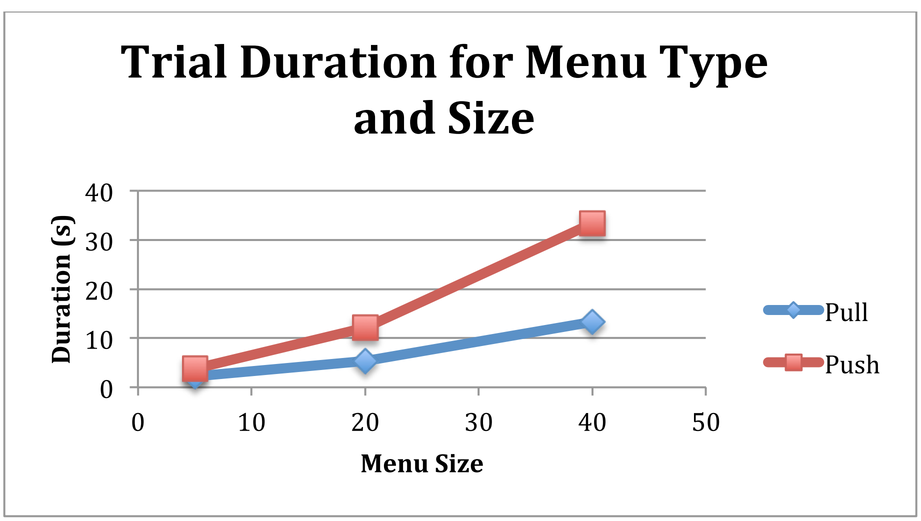

Figure 4.1: Menu Type * Menu Size Interaction on Trial Duration

The interaction is reasonable given that the pull menus allow preemption while the push menus require the participant to listen to the complete item (unless she selects it as the target). Therefore push menu selection durations should increase more than pull menus as menu size increases. Also, participants can better take advantage of learning a menu layout with pull menus in terms of time efficiency. In this case, a participant may know roughly where an item is in the menu and quickly advance to that general location in the menu before slowing down. This effect is likely amplified by longer menus.

Also, a between subjects analysis of the visual condition for main effects was conducted, but results were not significant.

A statistically significant interaction was detected between the visual condition and the menu type on selection duration, $F(1, 26) = 71.81$, $p < .05$, $\eta_p^2 = .73$, as well as the visual condition and the menu size on selection duration, $F(1.23, 32.06) = 47.71$, $p < .05$, $\eta_p^2 = .65$ (Huynh-Feldt sphericity correction). These three conditions showed a combined interaction on selection duration: $F(2, 52) = 42.32$, $p < .05$, $\eta_p^2 = .62$. These interactions are detailed in Figure 4.2 and descriptive statistics and pair-wise t-tests are detailed in Appendix B. All t-tests were significant (with Bonferroni correction) except when comparing trial duration of push menus of the same size across visual conditions.

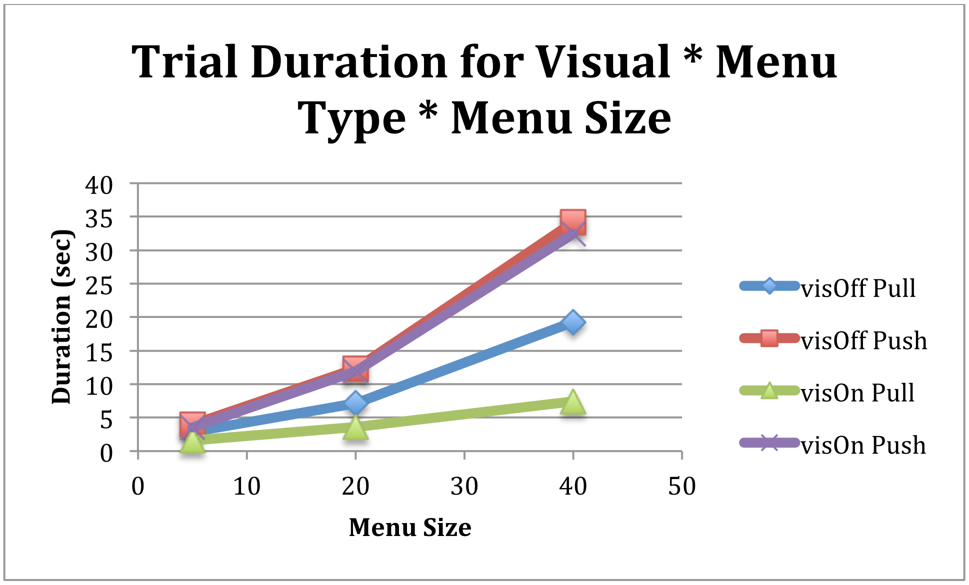

Figure 4.2: Interaction of visual condition * menu type * menu size on trial duration

These results show that pull menus are influenced by visual condition, correlating with improved target-seeking efficiency of participants; the difference becomes larger as menu size increases. However, push menus do not exhibit similar results.

Study 1 - Discussion

Overall, we can see that users are more efficient with the pull menu type and workload measures indicate the perception that push menus are more difficult to use. This is understandable, due to the longer period of engagement with the menu task for push menus. Menu item selection accuracy is quite similar for both menu types, which is promising given that push menus take longer to use.

The difference in the impact of the visual condition on menu type is hypothesized to be because participants cannot preemptively skip ahead to the target on push menus, even if they can see it on the visual menu list. However, with the pull menu type, this is possible. This effect is interesting from an interface design perspective. If a multimodal menu interface is used in a multitasking scenario (such as an infotainment system in an automobile), a user may be compelled to look at a corresponding visual representation of the menu to the detriment of the primary task of driving safely. A user may be less inclined to look at the screen with the push menu than the pull menu (or look less often and for shorter periods of time) due to the fact that there is little menu speed performance incentive.

Overall, this first study sets a baseline for future research in using push menus in multitasking scenarios. It is anticipated that the frustrations of push menus will be diminished in this more engaging context, yet push menus should less negatively impact a higher priority visual-motor task.

CHAPTER 5

USER STUDY 2 - PUSH AND PULL MENUS UNDER MULTITASKING CONDITIONS

After the first study largely confirmed hypotheses about the basic differences between push and pull menus, a new study was designed to explore the utility of push menus when a user's attention is divided. This new study utilized a primary task of a simple computer game in which the participant must catch balls falling from the top of the screen. Simultaneously, the participant searches for target items in a menu, very similar to the first study.

Several dependent variables were measured including menu selection accuracy, trial duration (time until selection made), game performance, workload, preference, etc.

Related to these variables, there were several hypothesized outcomes, many the same as with the first study. First, menu selection accuracy was expected to be worse and trial duration longer for the push menu type. Also, accuracy was expected to be worse, duration longer, and workload higher, as menu size increased for either menu type. The visuals-off selection accuracy was expected to be worse, trial duration longer, and workload worse than visuals-on. Game performance, however, was hypothesized to be better for the push menu type than the pull menu type. Also, perceived workload was hypothesized to be lower for the push menu type given the demands of the game task. In this study, the software was modified to deliver menu selection trials to the participant after a variable break of several seconds (randomly selected from 5, 10, or 15 seconds), but still continuously playing the game. This allowed segmenting the game play data into a *during-trial* and *off-trial* game accuracy. Game accuracy was hypothesized to be the same for all conditions when off trial.

Study 2 - Participants

Thirty-five undergraduate students (after one exclusion: 21 male, 14 female; mean age = 19, SD=1.5) participated in a study for credit in a psychology course at the Georgia Institute of Technology. Participants gave consent to participate in the study. All reported normal or corrected-to-normal vision and hearing. One participant was excluded due to data corruption of his/her log files.

Study 2 - Apparatus

The experiment was run on two Dell Optiplex 990's, Intel Core i5-2400 CPU @ 3.10 GHz, 4 GB Ram, 64-bit Windows 7 Professional. Both were connected to Dell P2210 LCDs @ 1024x768 (letterboxed to 4:3 aspect ratio). The middle screen, with the ball drop game, was configured to the same dimensions and aspect ratio. Two screens (left and right) were at the same height, the edges touching the middle screen's edges, and tilted inwards so that the screens were perpendicular to participants' estimated relative head position in an effort to optimize viewing angle. These were both connected to one computer, but only one was turned on corresponding to which hand was controlling the computer. The headphones from the previous study were used (Sennheiser HD202 Headphones). The Dell keyboard that came with the Optiplex was used to control the ball drop game. A Nintendo Wii Remote was used to control the experiment software and auditory menus (see Study 2 – Design and Procedure). The experiment was conducted in a sound proof booth in the Sonification Lab at Georgia Tech. The computers were outside of the booth and interfaced with peripherals via an access tunnel.

The ball drop game, discussed previously, was configured for automatic difficulty calibration with a target success rate of 85% (+/-5%). The participant performed the calibration step before the main study began.

Study 2 – Design and Procedure

Given that the second study was dual task, careful consideration was made to how the participant would make interface inputs to the game and auditory menu. It was decided to use a separate controller for each hand. The game is always controlled via arrows on a keyboard (left and right) sitting on a desk in front of the participant. The participant always controls the auditory menu with a handheld game controller. However, unlike the first study that used an Xbox 360 controller, a Nintendo Wii Remote is used instead. The reason for this change is that the Xbox 360 controller requires two hands while the Wii Remote only needs one. In the case of pull menus, the participant can move up or down to navigate the menu items with a thumb-controlled direction pad on the top of the Wii Remote. Selections are made with an index-finger trigger on the underside of the Wii Remote (labeled the "B" button). This decision was made (as opposed to using the "A" button on top of the controller for selection) so that the participant does not need to move her thumb off of the direction pad to make selections when performing under the pull menu condition. Push menus also use the "B" button on the Wii Remote to control for any affects of ergonomics, dexterity of fingers, etc.

Participants were also screened for handedness and randomly assigned to use their dominant hand for the game and other hand for the auditory menu, or vice versa. Three computer screens were placed on the desk in front of the user. The center screen always displays the game. The left and right screens are both interfaced with the computer running the auditory menu software. However only one is turned on such that the visuals are either to the left or right of the game and coinciding with the hand of the participant that is assigned to control the auditory menu.

Except for the dual task nature of Study 2, the design and procedure of the study were nearly the same as Study 1. In addition to menu training, there was also the game calibration step, and dual task training. Participants were randomly assigned to either train on the menus first or train on the ball drop game first. The last training step was always practice with the dual task scenario of playing the game and using the menu at the same time. Menu type and menu size conditions of paired blocks were randomized as described in Study 1 along with interleaved NASA TLX ratings. The study was fully automated as before. The visual condition was between subjects, and the condition was randomly assigned via coin toss.

Study 2 - Results

Results were analyzed with a 2 (push or pull menu type) x 3 (menu size: 5, 20, 40) repeated-measures analysis of variance (ANOVA). (Since the previous study showed no main effect or interactions between the block index of each condition, that analysis is omitted and the two blocks are treated as one.)

Statistically significant differences were found between menu types in mean accuracy of selection, $F(1,33) = 18.25$, $p < .05$, $\eta_p^2 = .36$ mean duration of trial, $F(1,33) = 151.37$, $p < .05$, $\eta_p^2 = .82$, and mean game accuracy, $F(1,33) = 20.46$, $p < .05$, $\eta_p^2 = .38$. Differences in menu types between mean perceived workload were not significant, $F(1,33) = 1.23$, $p = .28$, $\eta_p^2 = .036$.

Table 5.1: Descriptive Statistics

Menu Type	**Menu Accuracy**		**Duration**		**Game Accuracy**		**Workload**	
	Mean	**Std. Dev.**	**Mean**	**Std. Dev.**	**Mean**	**Std. Dev.**	**Mean**	**Std. Dev.**
Pull	.95	.07	10.77	2.72	.84	.1	51.86	18.29
Push	.90	.08	16.68	1.46	.88	.09	49.37	18.13

There were also statistically significant differences between menu sizes in mean accuracy of selection, $F(2,66) = 4.85$, $p < .05$, $\eta_p^2 = .13$, mean game accuracy, $F(2,66) = 51.05$, $p < .05$, $\eta_p^2 = .61$, mean trial duration, $F(1.27, 41.90) = 1057.51$, $p < .05$, $\eta_p^2 = .97$ (Huynh-Feldt sphericity correction), and mean workload, $F(1.56, 51.50) = 13.26$, $p < .05$, $\eta_p^2 = .29$ (Huynh-Feldt sphericity correction).

Paired-sample t-tests were conducted, with Bonferroni adjustment (3 comparisons), for the statistically significant main effect of menu size on menu selection accuracy, game accuracy, trial duration, and workload. Menu Size 5 menu selection accuracy ($M = .94$, $SD = 0.05$) was numerically (but not statistically significant) larger than Menu Size 20 menu selection accuracy ($M = .93$, $SD = 0.07$), $t(35) = 1.0$, $p = 1.0$, and was larger than Menu Size 40 menu selection accuracy ($M = .91$, $SD = 0.09$), $t(35) = 2.57$, $p < .05$. Menu Size 20 menu selection accuracy was numerically larger (but not statistically significant) than Menu Size 40 selection accuracy, $t(35) = 2.17$, $p = .08$. Menu Size 5 game accuracy ($M = .89$, $SD = 0.08$), was larger than Menu Size 20 game accuracy ($M = .85$, $SD = 0.09$), $t(35) = 6.00$, $p < .05$, and was larger than Menu Size 40 game accuracy ($M = .83$, $SD = 0.10$), $t(35) = 8.57$, $p < .05$. Also, Menu Size 20 game accuracy was larger than Menu Size 40 game accuracy, $t(35) = 4.00$, $p < .05$. Menu Size 5 trial duration ($M = 3.74$, $SD = 0.63$) was shorter than Menu Size 20 trial duration ($M = 10.62$, $SD = 1.60$), $t(35) = -29.67$, $p < .05$, and was shorter than Menu Size 40 trial duration ($M = 26.80$, $SD = 3.82$), $t(35) = -36.49$, $p < .05$. Also, Menu Size 20 selection duration was smaller than Menu Size 40, $t(35) = -27.66$, $p < .05$. Menu Size 5 workload ($M = 43.47$, $SD = 20.03$) was less than Menu Size 20 workload ($M = 52.03$, $SD = 19.01$), $t(35) = -5.15$, $p < .05$, and was less than Menu Size 40 workload ($M = 56.34$, $SD = 18.06$), $t(35) = -4.19$, $p < .05$. Also, Menu Size 20 workload was less than (but not statistically significant) Menu Size 40 workload, $t(35) = -1.60$, $p = .08$.

Menu type and menu size interacted with effects on game accuracy, $F(1.64, 54.06) = 9.64$, $p < .05$, $\eta_p^2 = .23$ (Huynh-Feldt sphericity correction), trial duration,

$F(1.19, 39.32) = 116.42$, $p < .05$, $\eta_p^2 = .78$ (Huynh-Feldt sphericity correction), and menu accuracy, $F(1.58, 52.24) = 5.14$, $p < .05$, $\eta_p^2 = .14$ (Huynh-Feldt sphericity correction). Descriptive statistics are shown for menu type and menu size for these measures in Table C.1 in Appendix C. Table C.2 shows Bonferroni-corrected (9 comparison) paired-sample t-tests of menu size measures for each menu type. Similarly, Table C.3 shows Bonferroni-corrected (9 comparison) paired-sample t-tests of menu type measures matched across menu size. Figures 5.1, 5.2, and 5.3 visualize these differences.

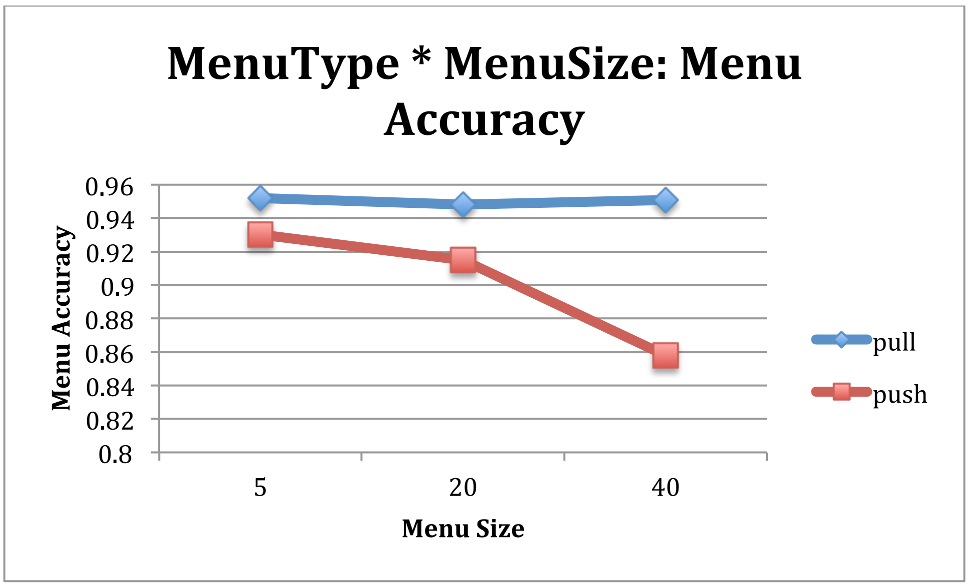

Figure 5.1: Menu Type * Menu Size on Menu Accuracy

One can see that menu accuracy for push menus drop well below pull once the menus get larger. Pull menus appear to hold relatively steady for menu accuracy across the different menu sizes. This effect might be explained by the fact that participants can easily correct overshooting their target with pull menus but not push menus.

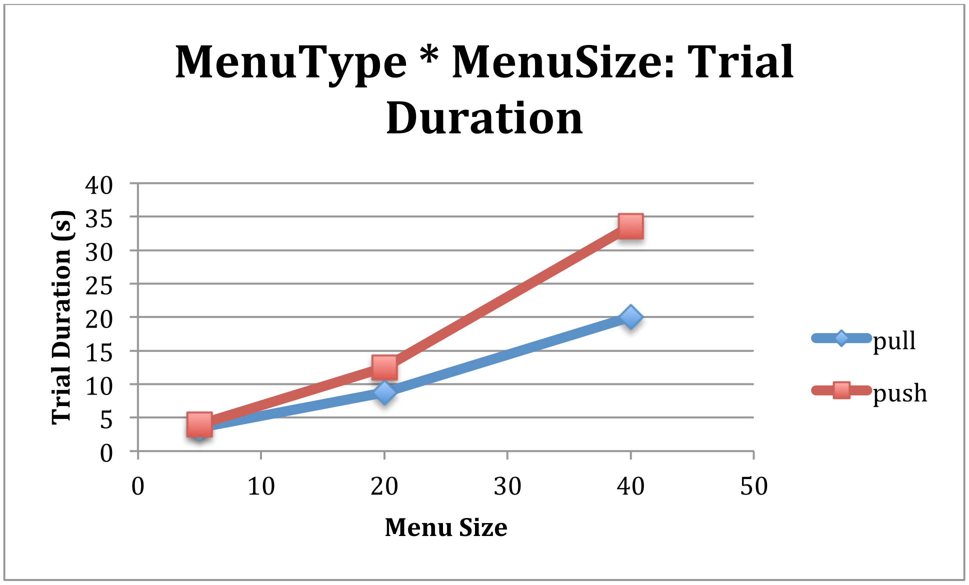

Figure 5.2: Menu Type * Menu Size on Trial Duration

Trial duration shows a very clear (and statistically significant) increase as menu size increases. Also, trial durations for each menu type are smaller for the pull type than the push type at each menu size.

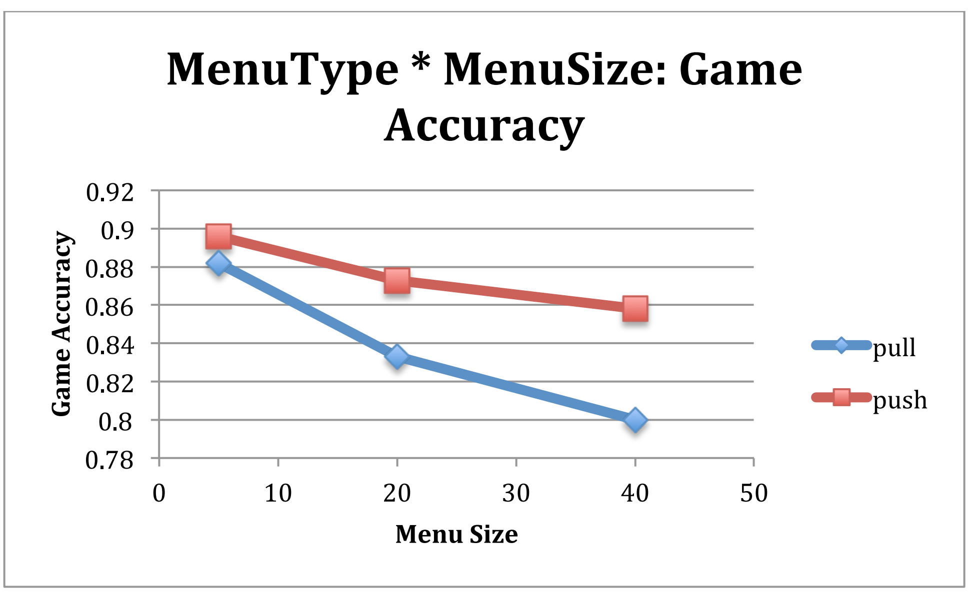

Figure 5.3: Menu Type * Menu Size on Game Accuracy

Game accuracy decreases (with statistical significance) as menu size increases for both menu types. Game accuracy is also lower for pull than push at each menu size, but only statistically significant for the two larger menu sizes.

A statistically significant interaction between menu type, and visual condition was found on trial duration, $F(1, 33) = 4.30$, $p < .05$, $\eta_p^2 = .12$. However, pair-wise t-tests with Bonferroni correction yielded no significant differences. The first study did show significant differences with similar conditions. Figure 5.4 seems to show a similar trend and may merit further study.

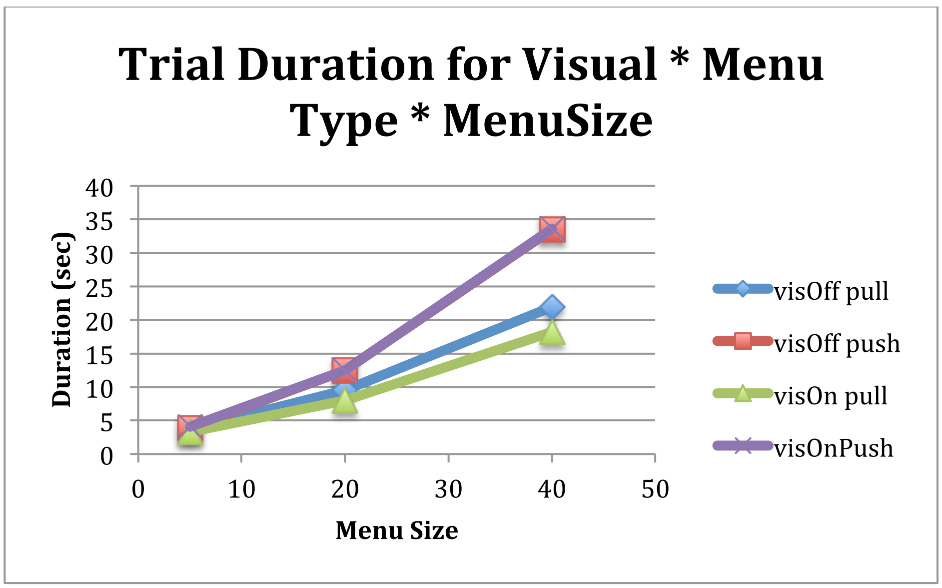

Figure 5.4: Visual * Menu Type * Menu Size on Trial Duration

Also, a statistically significant interaction between menu type, menu size, and visual condition was found on game accuracy, $F(1.64, 54.06) = 3.45$, $p < .05$, $\eta_p^2 = .10$. However, pair-wise t-tests with Bonferroni correction yielded no significant differences. Figure 5.5 hints at confounding effects of visuals-on with the pull menu type as menu size increases. This interaction looks to be important to observe in future studies. As previously discussed, participants are possibly more motivated to look at the screen with pull menus than push menus given the performance payoff that is possible for time to target menu item. Possibly, the longer the menu the longer that eyes are on the menu screen and therefore the primary task is more substantially affected.

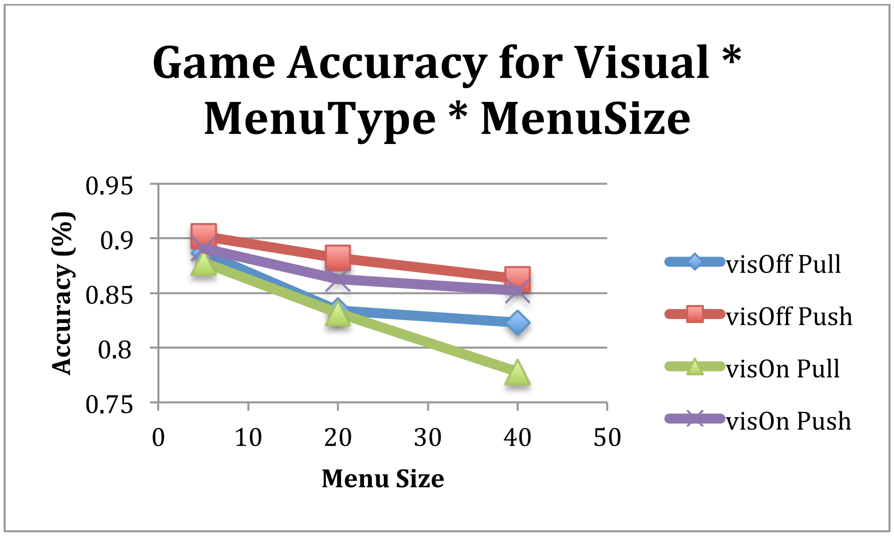

Figure 5.5: Visual Cond. * Menu Type * Menu Size on Game Accuracy

A second ANOVA was conducted with 2 (push or pull menu type) x 3 (menu size: 5, 20, 40) x 2 (trial state: during active trial, off-trial) repeated-measures analysis of ball drop game data. Only game accuracy is a meaningful measure with this segmentation of data. Trial state had a statistically significant effect on mean game accuracy, $F(1, 33) = 25.07$, $p < .05$, $\eta_p^2 = .43$.

Menu type and trial state had a significant interaction on game accuracy, $F(1, 33) = 43.40$, $p < .05$, $\eta_p^2 = .57$. Paired-sample t-tests were conducted, with Bonferroni adjustment (2 comparisons). Within the during-trial state, pull menu game accuracy (M = .78, SD = .15) was worse than push menu game accuracy (M = .88, SD = 0.07), $t(35) = -5.93$, $p < .05$. However, during the off-trial state, pull menu game accuracy (M = .87, SD = .10) did not have a statistically significant difference than the push menu game accuracy (M = .88, SD = .09), $t(35) = 0.14$, $p = 1.0$. For pull menus, game accuracy in the during-trial state was worse than the off-trial state, $t(35) = -5.81$, $p < .05$. There was no statistically significant difference in game accuracy for push menus across trial state, $t(35) = -0.60$, $p = 1.0$. Figure 5.6 shows visually the difference in game accuracy for pull and push menus across trial state.

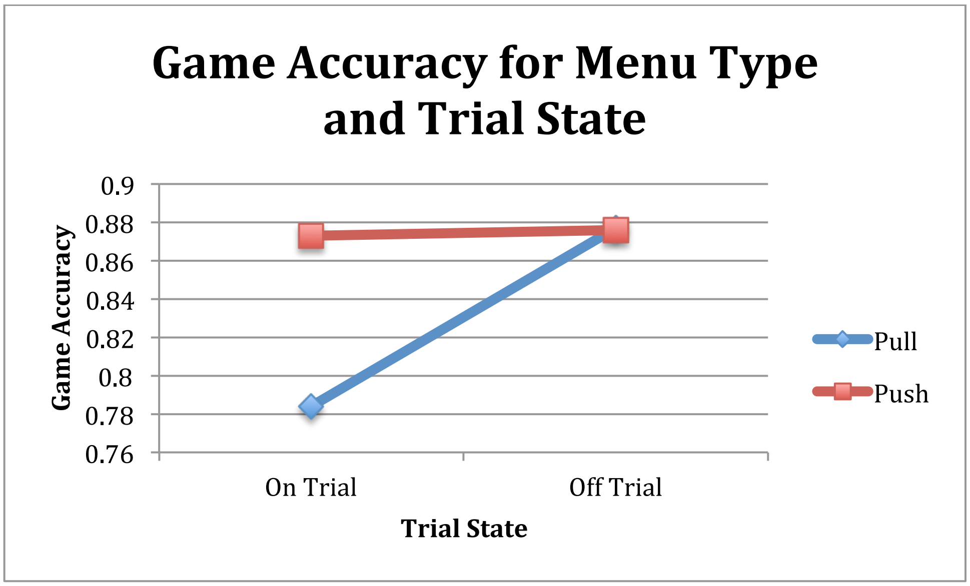

Figure 5.6: Game Accuracy for Menu Type and Trial State

When a participant is working on an active trial of the pull menu type, he/she must continuously interact with the menu. However, the push menu type only requires passive observation with no physical interactions until the selection is made. The simultaneous demands of tactile control for both tasks appear to have a negative impact on game performance.

Study 2 - Discussion

Study 2 demonstrates that push menus have a significant advantage over pull menus for dual task scenarios where a menu interface must be accessed by a user while simultaneously engaged with a demanding visual-motor primary task. This advantage is likely due to reduction of competing demands for the manual response stage of MRT.

Additionally, it is speculated that contention between manual tasks of controlling the game and simultaneously navigating the pull menu result in context switching that is costly for the game task. On the other hand, the push menu interaction keeps the user largely in the audio perceptual stage of MRT awaiting the target menu item. Therefore, concurrent multitasking is likely to be more effective.

To summarize the differences between push and pull menus observed in Study 2, push menus as compared to pull menus take longer to make selections and are more prone to inaccurate selections, and both differences are exacerbated by increase in menu size. However, pull menus have significant negative effects on demanding primary tasks, especially if the menu is longer and requires an extended period of time searching through the items. The menu performance suffers less because users can simply delay final selection until the target is found. Trial duration is more likely affected by the resource contention.

There also appears to be some possibility that a visual menu representation is more detrimental to pull menus than push menus. More research is necessary, but if confirmed, this may lead to design guidelines that suggest that visuals be removed from pull menu interfaces while users are operating vehicles. Also, push menus are probably better suited to simultaneous vehicle operation regardless to whether a visual menu is present.

CHAPTER 6

USER STUDY 3 - PUSH MENU OPTIMIZATION AND NAVIGATION CORRECTION

The third study went back to a single task and only looked at further exploring the push menu type and ways that push menus might be optimized to address the deficiencies of taking longer to make selections and inaccuracies in menu item selections as compared to pull menus. Dependent variables similar to the previous studies were measured including menu selection accuracy, trial duration (time until selection made), workload, preference, etc.

This third study did not further evaluate the pull menu type, there were instead two push menu types. First was the original push interaction, which is the same as the previous studies, but also a new mode called *push-with-correction.* This new push interaction was the same as the standard push except that it allowed the use of a second button. The second button allowed the user to back up one item in the auditory menu (preemptively). The hypothesis is that a user might realize that she has just missed the target item and can backup and then make the correct selection.

The experiment also controlled for menu sizes of five and twenty (using the corresponding content from the previous studies). The menu of size forty was dropped due to time constraints of the study.

Lastly, the silence interval between menu item presentation was controlled to selections of 200, 400, and 600 milliseconds.

Study 3 - Participants

Thirty-three (33) undergraduate students (17 male, 16 female; mean age = 19, SD=1.5) participated in a study for credit in a psychology course at the Georgia Institute of Technology. Participants gave consent to participate in the study. All reported normal or corrected-to-normal vision and hearing.

In Study 3, one participant was excluded due to software error and four participants showed signs of abandoning the study and intentionally entering the first possible answer on every trial. These four participants were excluded. Additionally, six participants were excluded for what was considered misuse of the correction feature. These participants' data shows that they would use the backup feature to go backwards through the menu very much like a push menu. The specific condition for exclusion was any participant rolling over from the first item backwards to the last item more than 10 times total was excluded. The number of times rolling over backwards for these six participants was 75, 48, 29, 27, 17, and 16. The data showed that no participant rolled over backwards more than once in a trial. The total number of trials in blocks with the push-with-correction type were 120.

After exclusions, twenty-two participants remained (12 male, 10 female; mean age = 19, SD = 1.2).

Study 3 - Apparatus

The experiment was run on a Dell Optiplex 990's, Intel Core i5-2400 CPU @ 3.10 GHz, 4 GB Ram, 64-bit Windows 7 Professional. The computer was connected to a Dell P2210 LCD @ 1024x768 (letterboxed to 4:3 aspect ratio). An Xbox360 controller was used for controlling the experiment software. The set up was nearly the same as Study 2, but the second computer and side screens were removed.

Study 3 – Design and Procedure

The design and procedure of Study 3 were essentially the same as Study 1, except there is no longer a between subjects visual condition. The different within-subject conditions of the push menu variants and silence intervals were randomized and then the two menu sizes randomized (2 blocks of 10 trials each, in a row for each overall condition).

Study 3 - Results

Results were analyzed with a 2 (push mode: basic or correctable) x 2 (menu size: 5, 20) x 3 (silence duration: 200 ms, 400 ms, 600 ms) repeated-measures analysis of variance (ANOVA).

Statistically significant differences were found with menu sizes in mean accuracy of selection, $F(1,21) = 25.98$, $p < .05$, $\eta_p^2 = .55$, mean duration of trial, $F(1,21) = 11308.13$, $p < .05$, $\eta_p^2 = 1.00$, and mean workload, $F(1,21) = 40.08$, $p < .05$, $\eta_p^2 = .66$.

Table 6.1 – Menu Size Effects Descriptive Statistics

Measure	Menu Size	Mean	Std. Dev.
Accuracy	5	.95	.04
	20	.93	.03
Duration	5	3.64	0.20
	20	11.45	0.45
Workload	5	25.89	12.59
	20	32.63	12.23

Also, statistically significant differences were found with silence duration in mean accuracy of selection, $F(1.32,27.80) = 21.65$, $p < .05$, $\eta_p^2 = .51$ (Huynh-Feldt sphericity correction), mean duration of trial, $F(2,42) = 656.10$, $p < .05$, $\eta_p^2 = .97$, and mean workload, $F(1.59,33.37) = 34.71$, $p < .05$, $\eta_p^2 = .62$ (Huynh-Feldt sphericity correction).

Paired-sample t-tests were conducted, with Bonferroni adjustment (3 comparisons), for the statistically significant main effect of silence duration on menu selection accuracy, trial duration, and workload. Silence duration 200ms selection accuracy ($M = .89$, $SD = 0.08$) was worse than silence duration 400ms menu selection accuracy ($M = .96$, $SD = 0.03$), $t(22) = -4.53$, $p < .05$, and was worse than silence duration 600ms menu selection accuracy ($M = .97$, $SD = 0.02$), $t(22) = -5.13$, $p < .05$. Also, silence duration 400ms menu selection accuracy was numerically (but not statistically significant) worse than 600ms menu selection accuracy, $t(22) = -2.00$, $p = .18$. Given these results, there appears to be a general trend towards decreased accuracy of target item selection as a push menu plays back more quickly due to the size of the silence interval.

Silence duration 200ms trial duration ($M = 6.40$, $SD = 0.39$) was shorter than silence duration 400ms trial duration ($M = 7.51$, $SD = 0.33$), $t(22) = -17.06$, $p < .05$, and was shorter than silence duration 600ms trial duration ($M = 8.71$, $SD = 0.30$), $t(22) = -33.49$, $p < .05$. Additionally, silence duration 400ms trial duration was shorter than silence duration 600ms trial duration, $t(22) = -21.09$, $p < .05$. These results are unsurprising given the straightforward increase in push menu playback speed as the silent gaps between items increase.

Silence duration 200ms workload ($M = 38.41$, $SD = 16.17$) was larger than silence duration 400ms workload ($M = 26.90$, $SD = 12.47$), $t(22) = 5.25$, $p < .05$, and was larger than silence duration 600ms workload ($M = 22.46$, $SD = 10.59$), $t(22) = 6.96$, $p < .05$. Also, silence duration 400ms workload was larger than 600ms workload, $t(22) = 3.45$, $p < .05$. This suggests that workload increases as a push menu plays through choices more quickly.

There was a statistically significant interaction between menu size and silence duration on selection accuracy, $F(2, 42) = 5.82$, $p < .05$, $\eta_p^2 = .22$, and trial duration, $F(1.69, 35.54) = 318.54$, $p < .05$, $\eta_p^2 = .94$ (Huynh-Feldt sphericity correction). Paired-sample t-tests were performed, with Bonferroni adjustment (9 comparisons) for these interactions, shown in Figures 6.1 and 6.2 as well as tables in Appendix D.

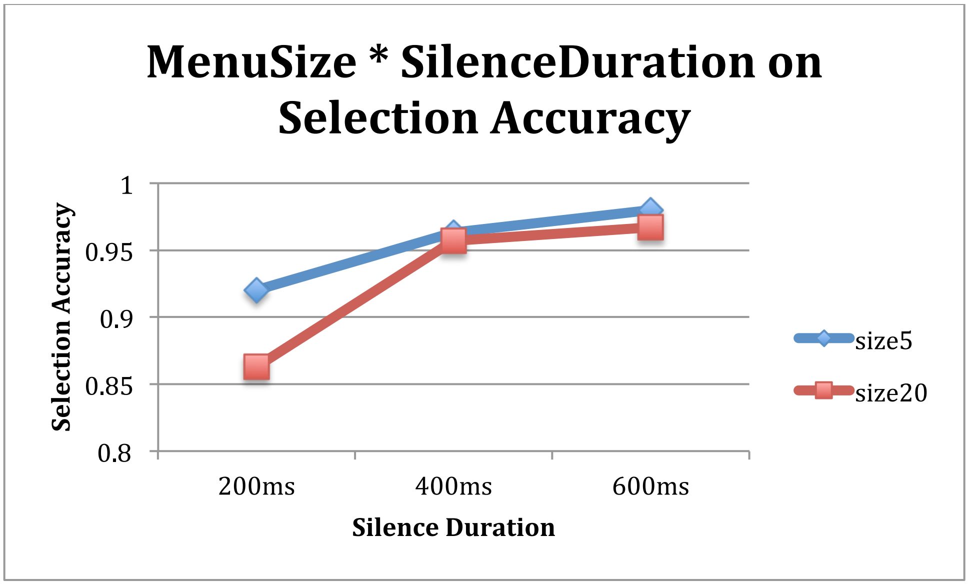

Figure 6.1: Menu Size * Silence Duration on Trial Accuracy

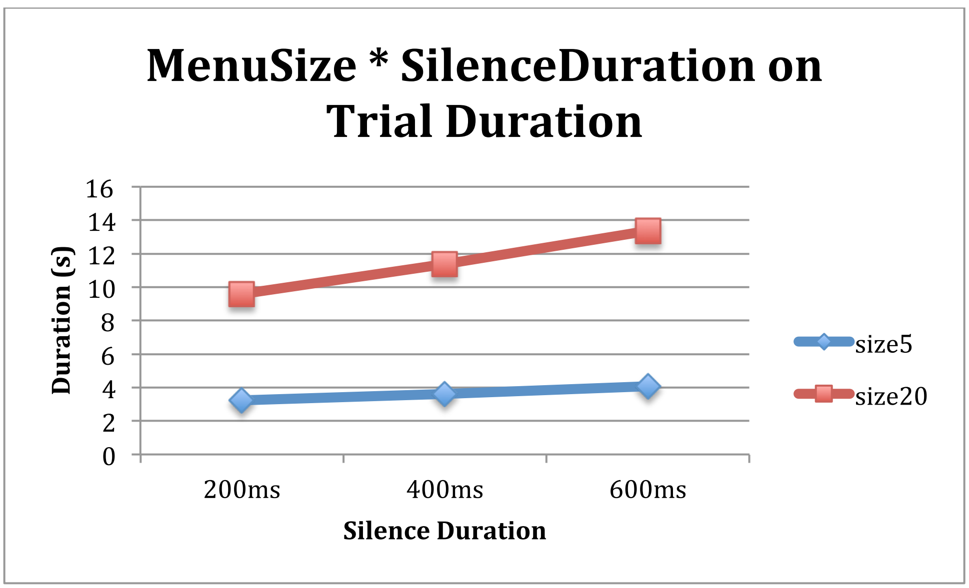

Figure 6.2: Menu Size * Silence Duration on Trial Duration

Within size 5 menus no statistically significant difference in the paired-sample t-tests are detected in selection accuracy. Size 20 menus however show the 200ms silence duration selection accuracy significantly lower. Furthermore, size 5 menu selection accuracy was significantly lower than size 20 at the 200ms silence duration. While all t-tests were not significant, it is hypothesized that if a similar experiment was conducted with more subjects that one would find a rapid drop-off in accuracy as silence duration decreases from larger durations down through a threshold between 200ms and 400ms. This drop-off would likely be exacerbated by menu size increase.

Trial duration shows a straightforward and intuitive difference over the menu sizes and silence durations. Since these are push menus, a decrease in silence duration will reduce the amount of time it takes to play the menu. Also, a longer menu takes longer to play through than a shorter menu.

Study 3 - Discussion

There appears to be a tradeoff between selection duration and accuracy for different silence durations between menu items in a push menu. On the one hand, selection duration can be reduced with smaller silence durations, but if the silence duration becomes too small then there is a significant drop-off in selection accuracy. The effect seems to be more substantial for both measures when menu size is larger.

This research does indicate that 400 ms silence duration is a reasonable compromise between the tradeoff mentioned above. Decreasing below this silence duration appears to have a rapid drop-off in performance (if interpolating between the 200 ms and 400 ms condition).

While the first attempt at addressing the ability of the user to make corrections to push menu interaction yielded no statistical results, it is interesting that several participants discovered an unanticipated use of the correction feature such that menu interaction effectively became a pull menu. This observation has interesting menu design implications. Users may be compelled to take advantage of any menu navigation features that they perceive as improving menu performance. This could be detrimental to primary task performance.

Therefore, it seems critical that menu designers should always be cautious in the introduction of any input features that augment menu interaction in some way. Any input that allows a change of direction, rate of speed, etc., may introduce the opportunity for a user to begin rapidly executing physical inputs with the menu system. This will likely introduce conflicting demands on the physical response stage of MRT. If the user is engaged with a critical visual-motor primary task, this primary task may be significantly impacted and the benefits of the push menu approach will be lost.

CHAPTER 7

USER STUDY 4 - FURTHER PUSH MENU OPTIMIZATION

Study 4 looks at another attempt to improve the efficiency of push menu interaction. This study alters the previous convention of push menu selection, which is that a selection for any particular item can be made during playback of that item or during the silence interval after the item yet before the next item. This new approach allows for a selection that is made during a later portion of the target item, during the silence interval after the target item, or during a portion of the beginning of the next item. The hypothesis is that a user will take some time to recognize an item and likely won't select during the first fractions of a second after playback begins. Instead, a selection during the very beginning of an item is likely meant for the previous item. This offers the opportunity to temporally compress the push menu, yet still provide a selection duration per item that is perceived as being essentially the same as with a menu without the overlap period.

This overlap concept was partially informed by previous work in spoken word recognition (Allopenna, Magnuson, & Tanenhaus, 1998). This studied spoken word recognition via eye tracking fixations on a 2D grid of objects represented by icons as identified by a spoken target object name. The actual task involved moving the target object via mouse drag-and-drop operation per a spoken instruction. Visual fixations identified the earliest point at which recognition can be acted upon via an individual. Aggregate data from this study identified about 400 ms from target offset (completion of target object name audio playback) to peak target fixation probability (e.g. participant eyes focused on matching object icon). Additionally, it took on average about 200 ms before participants moved their eyes from a neutral fixation point to scanning for the matching object icon. This suggests that there is a 200 ms period which can be used for overlap selection before a participant begins to be able to act on the currently playing menu item.

This previous work by Allopenna informed the hypothesis that 200 ms silence duration and 200 ms overlap duration would be an ideal setting. Furthermore, Allopenna's research reinforced results from Study 1-3 of this dissertation that identified 400 ms beyond the end of the target playback (silence interval) as ideal for the selection period.

For Study 4, it was hypothesized that improvements in selection duration could be achieved with little or no effect on selection accuracy so long as the overall selection duration (silence duration + overlap duration) remained the same. Dependent variables similar to the previous studies were measured including menu selection accuracy, trial duration (time until selection made), workload, preference, etc.

Like the Study 3, the Study 4 did not evaluate the pull menu type. The experiment also controlled for menu sizes of five and twenty (using the corresponding content from the previous studies).

An overall total selection interval of at least 400 milliseconds was maintained for all conditions. Even though a 200 ms was predicted to be ideal, selection overlap was either 0 ms, 200 ms, or 300 ms. This resulted in corresponding silence intervals of 400 ms, 200 ms, and 100 ms. This approach was made to help confirm the predicted ideal settings.

Study 4 - Participants

Twenty-six (26) undergraduate students (17 male, 16 female; mean age = 19, SD=1.5) participated in a study for credit in a psychology course at the Georgia Institute of Technology. Participants gave consent to participate in the study. All reported normal or corrected-to-normal vision and hearing.

Study 4 - Apparatus

The study apparatus was the same as Study 3.

Study 4 – Design and Procedure

The study design was essentially the same as previous studies. Conditions of silence overlap were randomized, then menu size was randomized, with two blocks for each overall condition. The experiment was fully automated.

Study 4 - Results

Results were analyzed with a 2 (menu size: 5, 20) x 3 (silence overlap: 300 ms, 200 ms, 0 ms) repeated-measures analysis of variance (ANOVA).

Statistically significant differences were found with menu sizes in mean accuracy of selection, $F(1,25) = 5.51$, $p < .05$, $\eta_p^2 = .18$, mean duration of trial, $F(1,25) = 11406.37$, $p < .05$, $\eta_p^2 = 1.00$, and mean workload, $F(1,25) = 21.45$, $p < .05$, $\eta_p^2 = .46$. Descriptive statistics are shown in Table 7.1. Accuracy is slightly higher for size 5 menus than size 20. Selection duration is longer for the longer menu. Also, workload is higher for the longer menu. All results are expected in light of the results from the previous studies.

Table 7.1: Menu Size Main Effects on Accuracy, Duration, Workload

Measure	Menu Size	Mean	Std. Dev.
Accuracy	size 5	.96	.03
	size 20	.94	.04
Duration	size 5	3.27	0.21
	size 20	9.90	0.41
Workload	size 5	28.42	16.17
	size 20	36.03	17.60

Also, statistically significant differences were found with silence overlap in mean duration of trial, $F(1.51,37.85) = 214.95$, $p < .05$, $\eta_p^2 = .90$ (Huynh-Feldt sphericity correction), and mean workload, $F(2,50) = 5.68$, $p < .05$, $\eta_p^2 = .19$. Selection accuracy was not shown to a have a statistically significant difference. Table 7.2 shows descriptive statistics for main effects of silence overlap on measures (selection accuracy is also included). Table E.1 in Appendix E shows the results of paired-sample t-tests. Larger selection overlaps had shorter duration than smaller overlaps for all measures. For workload, there was only a statistically significant difference between the 300 ms overlap and the 0 ms overlap (no overlap). Numerical differences in workload measures suggest that a hypothesis worth further study is that workload may increase with overlap.

Table 7.2 Selection Overlap Main Effect on Duration and Workload

Measure	Overlap	Mean	Std. Dev.
accuracy	300 ms	.94	.04
	200 ms	.95	.04
	0 ms	.95	.05
duration	300 ms	5.80	0.31
	200 ms	6.37	0.34
	0 ms	7.60	0.49
workload	300 ms	35.22	17.15
	200 ms	32.15	16.60
	0 ms	29.31	17.75

There was a statistically significant interaction between menu size and selection overlap on trial duration, $F(2, 50) = 121.83$, $p < .05$, $\eta_p^2 = .83$. Paired-sample t-tests were performed, with Bonferroni adjustment (9 comparisons) for these interactions. All comparisons were significant. These results reinforce results from the previous studies observing the increase in trial duration as push menu size increases. See Appendix E for descriptive statistics and Bonferroni-adjusted pair-wise t-tests.

Study 4 - Discussion

The most interesting result from this study is that overlap did not result in statistically different push menu selection accuracy between varying degrees of selection overlap. While it cannot be assumed that there is no effect at all, any effect from selection overlap is likely very small. Therefore, it appears that use of overlap is a viable strategy for increasing push menu efficiency due to the fact that it decreases the amount of time it takes to play through the menu. Future work may be useful to determine if overlaps become detrimental for familiar menus. It is hypothesized that if a user is very familiar with a push menu, the user may attempt to anticipate the beginning of the target item and actually select during the overlap period. This suggests that a push menu system may need to adapt selection interval and overlap to a user as she learns static (or rarely changing) menus. However, there are design challenges regarding how to measure user error so as to inform adaptation of push menu configuration. Additionally, detection of ambiguous selections (e.g. temporally near a transition between menu items) may be used to trigger mediation techniques to refine selection. However, menu designers must be cognizant of the implications of further manual inputs, especially within a small time frame.

With the completion of Study 4, the utility of the push menu concept demonstrates a relatively broad range of configurable dimensions which can be optimized for a specific task or tailored to individual user ability. Push menus may require more thoughtful design and configuration than a pull menu, but once configured optimally the benefits in multitasking scenarios can be fully leveraged.

CHAPTER 8

USER STUDY 5 - PUSH AND PULL MENUS WITH DRIVING SIMULATION TASK

A final study was conducted, which is similar to the second study that looked at a dual task scenario with auditory menu interaction while playing a demanding visual-motor game. Instead of a game, the user performs a driving task in a driving simulator (OpenDS with a large screen and steering wheel with pedals).

Specifically, the user performs the driving with a Three Vehicle Platoon Task (3VP) that involves pacing a lead car that periodically performs braking. Additionally, a second car follows behind the user's car and is in view of the rear view mirror. The user must watch for the lead car's braking events and slow down without colliding. Also, the user must monitor the rear car in the rear view mirror and press a response button as soon as the user notices that the rear car turns on its turn signal.

It is noted that the 3VP task does have limitations in creating a realistic driving scenario. Participants are not performing a normal driving activity such as navigating to a location, interacting with traffic controls, sharing the road with other drivers exhibiting real world behavior, etc. Therefore, higher level impacts on situation awareness cannot be effectively measured by 3VP task performance. Furthermore, the 3VP task as provided by OpenDS involves driving on a perfectly straight road. This significantly diminishes the challenge of lane holding (lateral lane position deviation). However, the 3VP task has the advantage of being highly standardized and easy to control the rate at which stimuli are presented to the participant. This ability allows experiment designs such that 3VP stimuli are very likely to occur during secondary tasks such as auditory menu operations.

This study returns to comparison of push and pull menu types, also including a within subjects comparison of the visual condition of the menu (visuals on or visuals off).

A medium size menu of twenty items is used for all conditions, using the same content as the size twenty menu from previous studies. The push menu type is configured to use settings deemed optimal from the previous studies. Specifically, a silence and overlap interval of 200 ms for both metrics was used.

Participants were instrumented with an eye tracking system as well as continuous biometric measurement of heart rate and galvanic skin response (GSR).
Like the previous studies, NASA TLX workload assessment is measured between conditions.

Several dependent variables were measured including menu selection accuracy, trial duration (time until selection made), estimate of workload (NASATLX), gaze direction, number of glances, heart rate, galvanic skin response (GSR), etc. The instrument for measuring heart rate and GSR was unfortunately considerably problematic and ultimately that data was excluded from analysis.

The results were hypothesized to indicate similar effects as the first of the dual-task studies (Study 2 overall). In particular, the primary driving task performance was hypothesized to be impacted less for push menus than pull menus. Additionally, biometric measures were anticipated to indicate higher workload for pull menus relative to push. Unfortunately, technical issues resulted in unusable biometric measures (discussed further below).

Study 5 - Participants

Thirty-five participants (24 males and 11 females, mean age: 20.1) from Georgia Tech took part in the study. The average years with a driver's license was 4.0 years. Participants received research credit for participation. All participants were required to be at least 18 years of age, have a valid drivers license for at least 2 years, and have normal or corrected-to-normal vision and hearing. One participant was excluded due to repeated wrecking of the simulated car with the lead car, resulting in passing of the lead car from which could not be recovered. The only fix was to restart the simulation software. No other participants wrecked the simulated vehicle.

Study 5 – Apparatus

The experiment software (with push/pull menu) was run on a Dell Optiplex 990's, Intel Core i5-2400 CPU @ 3.10 GHz, 4 GB Ram, 64-bit Windows 7 Professional. The computer was connected to a Lilliput 6"x4" screen that served as a center console. A Wiimote controller was used for controlling the experiment software, using the same button mappings as Experiment Two.

The OpenDS driving simulator was also run on a Dell Optiplex 990. It was connected to Logitech Driving Force GT Racing Wheel with floor pedals. The Wiimote was attached to the steering wheel via hook and loop straps and a custom molded brace made from cured shapeable silicone putty. A counter balance weight was attached to the opposite side of the steering wheel to balance the steering wheel.

Figure 8.1: Overview of experiment setup

Figure 8.2 Wiimote attached to steering wheel with counterweight

The overall physical layout consisted of a fixed chair 18.5” high and a table 29.5” high. The front of the chair was initially 7” from the table edge, but was sometimes adjusted slightly to accommodate users. The steering wheel of 11” diameter was about 14”/15” from the participant’s chest. The driving simulation screen (Samsung LCD HDTV, 1080P, 40” diagonal) was approximately 35” from the participant’s eyes. The center console screen was 24” to the right of the centerline, and 30” down the centerline, and approximately 15” down from eye height (dependent on participant height). The direct view distance was approximately 41”. The center console screen was oriented to be approximately orthogonal to the view angle. The center of the eye tracking system was approximately 30” away.

The FaceLAB eye tracking system was used to track eye direction. It uses reflected infrared light and stereoscopic cameras for tracking the eyes. It collects data at 60 Hz. The software was running on a Dell Latitude D830 laptop. The eye tracking system can also track pupil size. However, it was discovered that accurate pupil tracking generally only works well with participants with light colored eyes and not wearing contacts. Otherwise, iris tracking worked better for gaze tracking (e.g. for dark colored eyes).

Biometric measures of heart rate and galvanic skin response were collected with a NeXus-10 physiological monitoring and feedback platform connected to an Acer Aspire 5750-6421 laptop, with recordings made at 32 Hz. The heart rate electrode pads were placed in the modified lead II configuration. Specifically, the layout was with the positive lead on the left lower ribs, the negative lead under the right clavicle, and the ground under the left clavicle.

All computers were synced to Network Time Protocol (NTP) a time server located at Georgia Tech and synchronized via the NetTime program (http://www.timesynctool.com). This allowed data collection logs to be easily compared across systems.

Study 5 – Design and Procedure

This study protocol was very similar to the second study. Participants were tasked with finding menu items in a list of twenty items while simultaneously performing the OpenDS driving simulator 3VP task.

A 3VP script generator was developed specifically for this study that generates a randomized event schedule with a modified Poisson distribution. The modification is that a minimum and maximum time between consecutive events is enforced. At each event point, a random number and configurable ratio determines whether the event will be braking of the lead car or turn signal of the rear car. An additional accumulating bias corrects event type distribution back towards expected means quicker than chance alone (to avoid long strings of the same stimulus event). Overall, the script generator creates a semi-random script, ensuring that event type and event schedule are unpredictable, but still produce good coverage, especially including 3VP events that often occur simultaneously with menu tasks.

After informed consent was obtained, participants were instructed with how to instrument themselves with the heart rate sensors in a private dressing room. Additionally, participants were instrumented with a GSR finger sensor. Next they performed eye-tracking calibration and then received automated menu interaction training. This was followed by driving task training, and then dual task (driving plus menu) training. After all setup and training, participants completed two paired blocks each finding ten binned targets of the menu list (ultimately covering all list items) across the randomized conditions of visual condition of the menu (on or off), and menu type (push or pull). Unlike the second study, it was not practical to automate the start and stop of all computer systems. Therefore, the experiment administrator manually completed this procedure and correlated time stamps in logs from each of the computers to compensate for the differences (network time synchronization was utilized).

Some log files did not log high-resolution absolute time stamps for all samples and instead only recorded the absolute time at the start and then either relative offsets for sample times or frame counts. The worst case for comparison purposes was the eye tracker log files that only recorded an absolute time stamp for the start of collection and had an accuracy of seconds rather than milliseconds. This means that it was impossible to correlate events across log files as precisely as was done in the second study (only to one second accuracy).

There was a delay of 5, 10, or 15 seconds between each trial (as before). As with the second study, this allowed for the creation of a condition representing on/off trial status. Data logs were segmented into samples within the period of finding a menu target or the period of waiting for the next target.

Measures were recorded from OpenDS 3VP task including follow car turn signal response, lead car speed reduction task, vehicle speed, lateral lane position, lateral lane position variation (an approximate measure of lane position corrections, further discussed below), lateral lane deviation, lead car follow distance, and lateral lane deviation. The deviation measures specify a distance outside of an acceptably range defined by the OpenDS system. If a participant operates her vehicle within the acceptable range then a value of zero is logged. Otherwise the distance from the edge of the range is logged. OpenDS records logs to a database at approximately 15 Hz.

A new variable was created from the OpenDS logs. This was a simple measure of lateral lane position variations meant to capture magnitude of user corrections to lane drift over time. Unfortunately OpenDS does not have direct logging of steering wheel inputs. To accomplish this, the standard deviation of lane position was calculated. This approach was assumed to be effective because the sampling rate was uniform (at 15 Hz) and directional changes only occur from steering inputs (no simulated wind, programmatic lane drift, etc.).

The eye tracker records gaze direction, pupil size, and gaze object (determined by a scene geometry configuration). The scene in this study includes a model of the simulator screen relative to the eye tracker cameras. This allows determination of whether a user is looking at the screen or not. The center console screen was not modeled as it was found to be too small to accurately determine when it was looking at. Pupil size was ultimately deemed to be problematic because the ability to track pupil size was highly variable across participants (eye color, contacts, etc.). This is further complicated by the fact that the FaceLAB system can only be configured for pupil-based or iris-based gaze tracking. Most often the iris-based tracking worked best for gaze direction, leaving very few pupil-tracked participants. Ultimately, the pupil size variable was discarded due to lack of statistically viable data.

Also, a new variable was created from the eye tracker logs. This was a count of the number of times the participant glanced away from the driving simulator screen. This was accomplished by analyzing individual frames of log data looking for a transition from looking at the simulator to not looking at it and back again, marking events at each transition. The events were filtered to remove glances away shorter than 10 ms (assumed to be intermittent tracker loss rather than an actual glance). Then the sum total was reported as the number of glances away.

Study 5 – Results

Results were analyzed similarly to Study 2. Measures such as driving performance and eye tracking measures were divided into *on trial* and *off trial* status groups. The *on trial* category refers to measures taken during an auditory menu selection task. The *off trial* category refers to measures taken during the break between trials, but still during an experiment block. This design results in a 2 (menu type: push, pull) x 2 (visual condition: visuals on, visuals off) x 2 (trial status: on, off) repeated-measures analysis of variance (ANOVA) for driving and eye tracking measures. For menu interactions and NASATLX workload, a 2 (menu type: push, pull) x 2 (visual condition: visuals on, visuals off) repeated-measures analysis of variance (ANOVA) is applied. By chance, two participants did not experience at least one turn signal or at least one speed reduction stimuli during one of the on/off trial periods. These two participants were excluded for OpenDS measures analysis. Therefore all driving performance metrics use an N of 32. Note that after exclusions, the mean number of turn signal events per on/off trial period was 7.1 (*SD* = 2.4, *Min* = 1, *Max* = 15), and the mean number of speed reduction events per on/off trial period was 7.1 (*SD* = 2.2, *Min* = 1, *Max* = 13).

Statistically significant differences were found with menu type in delay of response to the turn signal stimulus of the 3VP task, $F(1, 31) = 4.29$, $p < .05$, $\eta_p^2 = 0.12$.

Table 8.1 Menu Type Main Effects on OpenDS 3VP Event Responses

	3VP Turn Sig Resp. (ms)		3VP Speed Red. Resp (ms)	
MenuType	**Mean**	**Std. Dev.**	**Mean**	**Std. Dev.**
Push	1181.31	232.29	1194.47	145.95
Pull	1226.53	240.77	1191.53	103.54

The response to turn signal is 45ms quicker for the push menu than the pull menu. This could be an influence of the visual on condition with the pull menu, but that interaction was not significant. Another possibility is differences in workload (and resource contention) for the two tasks. Interestingly, the speed reduction response time mean was numerically very close between the two menu types, as seen in the table above. Given that failure to comply with the speed reduction would result in a simulated vehicle collision, participants likely placed the highest priority on this task in relation to the turn signal response and menu interactions. Therefore, it is hypothesized that lower priority tasks suffer sooner than higher priority tasks as resources are more heavily utilized. Lastly, the turn signal indicator in the rear view fills a roughly constant visual space and the location does not change much either. However, in addition to the brake lights of the lead car during a speed reduction event, there is also a change in perspective as the lead car gets closer. Visually, the lead car gets bigger as the speed reduction occurs unchecked. This fact may also contribute to a quicker response relative to the turn signal. The turn signal stimulus is actually somewhat similar to the Peripheral Detection Task (PDT) (Martens & Van Winsum, 2000), which has known characteristics in response to workload. The speed reduction stimulus is arguably more distinct from PDT, and that perhaps explain some of the observed difference in response time.

Statistically significant differences were found with the visual condition on turn signal response stimulus of the 3VP task, $F(1, 31) = 4.65$, $p < .05$, $\eta_p^2 = 0.13$, and the speed reduction response stimulus (3VP), $F(1, 31) = 5.72$, $p < .05$, $\eta_p^2 =0.16$.

Table 8.2 Visual Condition Main Effects on 3VP Event Responses

Vis. Cond.	3VP Turn Sig Resp. (ms)		3VP Speed Red. Resp. (ms)	
	Mean	Std. Dev.	Mean	Std. Dev.
On	1229.21	241.60	1215.69	132.42
Off	1178.63	233.98	1169.98	119.85

The visuals on condition resulted in 50.57 ms mean increase in response time for the turn signal stimuli and 45.71 ms mean increase for the speed reduction stimuli. Again, these are expected outcomes if participants are looking away from the road when the stimuli occur.

The on/off trial grouping had a statistically significant effect on vehicle speed, $F(1, 31) = 20.30$, $p < .05$, $\eta_p^2 = 0.40$, lateral lane position, $F(1, 31) = 7.29$, $p < .05$, $\eta_p^2 = 0.19$, and the distance to the lead car, $F(1, 31) = 9.78$, $p < .05$, $\eta_p^2 = 0.24$.

Table 8.3 Trial Status Main Effects on OpenDS Measures

TrialStatus	Speed (kph)		Lane Pos. (m)		Lead Car Dist. (m)	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
On	54.25	5.36	0.30	0.20	22.01	2.79
Off	54.99	5.42	0.29	0.20	21.80	2.82

Participants' speed was slightly higher off trial. A possibility is that participants slow down during trial as a function of resource contention or increased workload. Lane position was slightly closer to the lane centerline when off trial; perhaps suggesting on trial menu interactions with menu buttons caused a drift to the right (menu buttons on the right of the steering wheel). Another possibility is that further from the centerline was considered a safer place to escape from pending collision and participants were aware of the diminished ability to focus on the primary driving task. Also, participants maintained their car closer to the lead car when off trial. Again, resource contention could be the cause of slowing down when on trial, and falling behind the lead car.

A statistically significant interaction of menu type on on/off trial grouping was detected with longitudinal deviation, $F(1, 31) = 4.56$, $p < .05$, $\eta_p^2 = 0.13$. However, post hoc t-tests showed no significant results.

There was a statistically significant interaction of visual condition with the on/off trial grouping on the 3VP turn signal response, $F(1, 31) = 4.87$, $p < .05$, $\eta_p^2 = 0.14$, the 3VP speed reduction response, $F(1, 31) = 5.50$, $p < .05$, $\eta_p^2 = 0.15$, and on lateral lane movement, $F(1, 31) = 5.40$, $p < .05$, $\eta_p^2 = 0.15$.

Post hoc t-tests on the turn signal response affected by visual condition and on/off trial status were not significant after applying a Bonferroni correction of 4. Post hoc t-tests on lateral lane movement for the same interaction did not yield significant results either.

Post hoc t-tests on the speed reduction response (by visual plus on/off trial status interaction) were only significant for comparisons between the visual on condition when on trial ($M = 1247.89$, $SD = 30.72$) and visual on condition when off trial ($M = 1183.49$, $SD = 23.59$), $t(31) = 2.71$, $p < .05$ (Bonferroni of 4). Again this fits with the possibility of increased resource contention causing delayed response. That this only affected the visual on condition suggests that there is increased workload caused by including visual interface elements.

Eye tracking measures had several exclusions identified during collection due to clear loss of tracking or clearly incorrect gaze direction as viewed in the live preview screen of FaceLAB. A total of eight participants were excluded due to these observed eye-tracking problems. This resulted in an N of 26 for the following analysis of eye tracking measures.

The visual menu condition had main effects on the percentage of time participants spend looking at the driving simulator computer screen, $F(1, 25) = 5.87$, $p < .05$, $\eta_p^2 =$

0.19, as well as the number of glances away from the simulator, $F(1, 25) = 17.54$, $p < .05$, $\eta_p^2 = 0.41$.

Table 8.4: Visual Condition Main Effects on Eye Tracker Measures

VisCond	Eyes on Driving Sim %		Num. Glances Off Screen	
	Mean	Std. Dev	Mean	Std. Dev.
On	0.969	0.001	14.38	11.88
Off	0.985	0.001	6.07	9.25

When the visual component of the menu is turned off, participants spend more time looking at the driving simulator screen. Note that participants may still occasionally look at the center console screen if they forget what their menu target is (the target is always displayed during a trial) or if participants are anticipating the end of a block. In retrospect, the lack of an auditory alert as to the end of a block was an experiment design mistake, even with the experiment administrator notifying participants when blocks were complete. This observation is further addressed in the interaction of visual condition and on/off trial status discussed later in the results.

The on/off trial grouping had main effects on the percentage of time looking at the simulator screen as well, $F(1, 25) = 8.06$, $p < .05$, $\eta_p^2 = 0.24$.

Table 8.5: Trial Status Main Effect on Eye Tracker Measure

Trial Status	Eyes on Driving Sim %	
	Mean	Std. Dev.
On	0.972	0.001
Off	0.982	0.001

While participants are interacting with the menu tasks, they spend less time looking at the driving simulator screen. This is probably most likely influenced by the visual condition, which will be observed in the following interactions.

A statistically significant interaction of the visual condition of the menu with the on/off trial grouping condition on percentage time looking at the driving simulator screen was observed, $F(1, 25) = 14.67$, $p < .05$, $\eta_p^2 = 0.37$. Post hoc t-tests were performed and statistically significant differences were found between on-trial/visuals-on ($M = 0.957$, $SD = 0.046$) and on-trial/visuals-off ($M = 0.987$, $SD = 0.025$), $t(25) = 3.00$, $p < .05$, as well as on-trial/visuals-on and off-trial/visuals-on ($M = 0.981$, $SD = 0.020$), $t(25) = 3.43$, $p < .05$ (Bonferroni correction of 4). Off-trial/visuals-off ($M = 0.984$, $SD = 0.025$) was not significantly different than off-trial/visuals-on. Likewise, when visuals are off, on/off trial status shows no statistically significant difference for the simulator view percentage. Otherwise, a reduction in screen viewing percentage is observed when participants transition from off-trial to on-trial when in the visuals-on menu condition, and a similar reduction is observed when going from visuals-off to visuals-on while on trial.

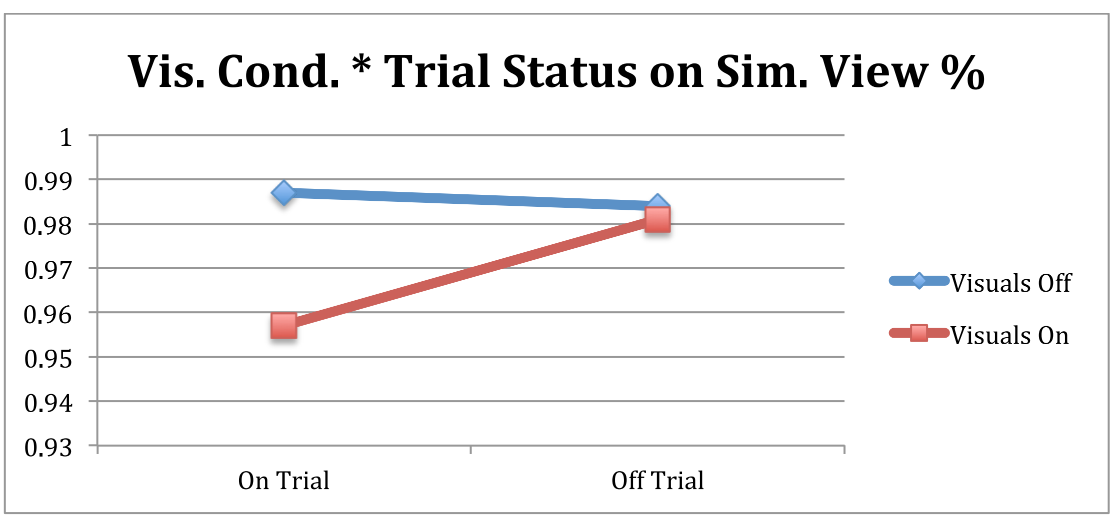

Figure 8.3: Visual Condition Interaction with Trial Status on Eye Tracker Measure

There was also an interaction of the visual condition with on/off trial status on the number of glances away from the simulator screen, $F(1, 25) = 26.00$, $p < .05$, $\eta_p^2 = 0.39$. Post hoc t-tests were performed and visuals-on/on-trial ($M = 17.75$, $SD = 16.14$) was significantly different than visuals-off/on-trial ($M = 4.69$, $SD = 7.82$), $t(25) = 4.41$, $p < .05$, visuals-on/on-trial was significantly different than visuals-on/off-trial ($M = 11.00$, $SD = 10.11$), $t(25) = 2.71$, $p < .05$, and visuals-off/on-trial was significantly different than visuals-off/off-trial ($M = 7.44$, $SD = 10.91$), $t(25) = 3.31$, $p < .05$. (All t-tests Bonferroni corrected for 4 comparisons.) Visuals-on/off-trial and visuals-off/off-trial were not significantly different.

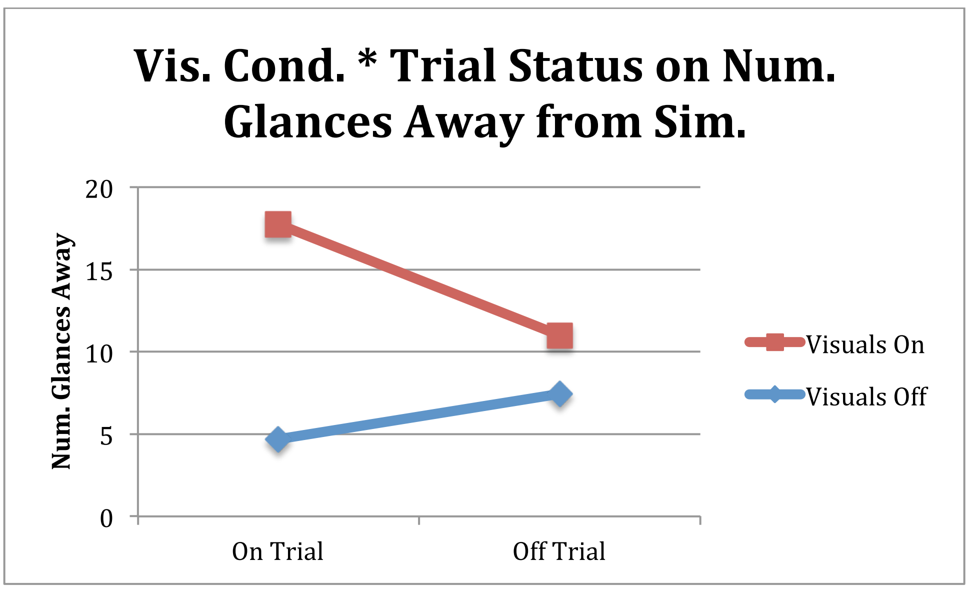

Figure 8.4: Visual Condition Interaction with Trial Status on Eye Tracker Glances

Collectively, these observations of the visual condition interaction on on/off trial status suggest that participants' percentage view of the simulator (or number of glances) is primarily negatively affected while on-trial and the visual menu is on. The mean number of glances in the visuals-off/off-trial are statistically significantly higher than visuals-off/on-trial. This does fit with the hypothesis that participants spent time quickly checking the center console display near the end of blocks to see if the block is complete. Also, while visuals-on/off-trial and visuals-off/off-trial were not statistically significantly different, numerical values do suggest that participants spent slightly more time looking at the center console in the visual-on condition. This is reasonable given that participants could review the visual menu list while waiting for a new trial.

The menu performance metrics cannot be meaningfully segmented into on/off trial groupings because menu interactions only take place on trial. Because of this, there were no issues with missing or excluded data, other than the one participant excluded overall, discussed previously. Therefore, analysis takes place on 34 participants ($N = 34$). Menu type had statistically significant main effects on the menu performance metrics of trial accuracy, $F(1, 33) = 80.22$, $p < .05$, $\eta_p^2 = 0.71$, and menu trial duration, $F(1, 33) = 10.68$, $p < .05$, $\eta_p^2 = 0.24$, as well as NASA TLX workload estimate, $F(1, 33) = 12.41$, $p < .05$, $\eta_p^2 = 0.27$.

Table 8.6: Menu Type Main Effects on Menu Performance and Workload

Menu Type	Menu Accuracy		Menu Duration		NASATLX	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Push	0.90	0.05	10.21	0.80	56.69	13.56
Pull	0.99	0.02	8.68	2.94	51.31	14.41

These results agree with previous studies with pull menus being superior in selection accuracy and trial duration, as well as lower perceived workload.

The menu visual condition had statistically significant main effects on menu trial duration, $F(1, 33) = 7.41$, $p < .05$, $\eta_p^2 = 0.27$.

Table 8.7: Visual Condition Main Effects on Menu Duration

	Menu Duration	
Vis. Cond.	Mean	Std. Dev.
On	9.18	1.86
Off	9.72	1.68

The increase in menu trial duration for the visuals off condition relative to visuals on is in agreement with the results of the first study that showed that pull menu interaction performance is increased by the presence of a visual menu. The same interaction has been observed in Study 5 and is detailed below.

A statistically significant interaction between menu type and menu visual condition was observed on menu trial duration, $F(1, 33) = 10.02$, $p < .05$, $\eta_p^2 = 0.23$. Post hoc t-tests were performed and statistically significant differences were found between push-menu/visuals-on ($M = 10.21$, $SD = 0.82$) and pull-menu/visuals-on ($M = 8.14$, $SD = 3.35$), $t(34) = 3.84$, $p < .05$, and between pull-menu/visuals-on and pull-menu/visuals-off ($M = 9.23$, $SD = 2.84$), $t(34) = 3.21$, $p < .05$ (Bonferroni correction for 4 comparisons). This result mirrors the interaction observed in the first study.

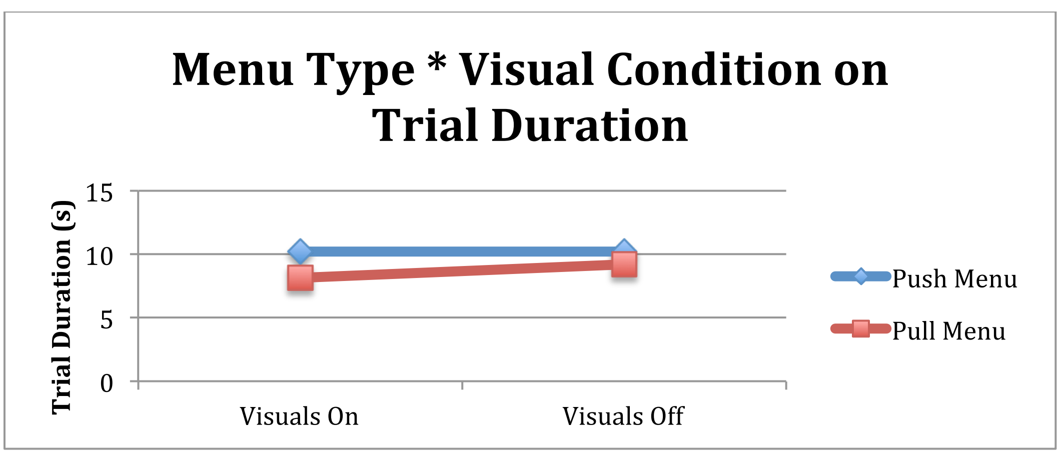

Figure 8.5: Menu Type Interaction with Visual Condition on Menu Trial Duration

Study 5 – Discussion

This study has provided evidence that push menus can have benefits over pull menus while an individual is simultaneously performing driving tasks. There is clear support for the hypothesis that push menus impact driving task less than pull menus, specifically regarding the turn signal response of the 3VP scenario. Users may be able to prioritize tasks overall such that activities with the most immediate potential for repercussions are managed (such as not running into the car in front of the driver). The impact of differing menu interactions on these critical tasks may be less distinct than the impact on supplementary diligence tasks like periodically checking rear and side view mirrors. This suggests that a future research direction could focus more on the impact of menu type on situation awareness in driving, or at least measures of driving performance deemed to be most impacted by the level of situation awareness. Again, it is noted that the 3VP task is limited in its ability to recreate a realistic driving scenario.

The simulated 3VP driving task has limitations in that it measures stimulus responses for events that occur at an unrealistic schedule. Additionally, the artificial nature of the 3VP scenario does not provide a real world scenario with higher-level tasks and opportunities for the user to demonstrate problem solving that may be affected by situation awareness. Furthermore, the 3VP task takes place on a simulated road that is perfectly straight. This likely results in a ceiling effect on lateral lane holding. Ideally, the road should involve some randomly determined curvature. It's hypothesized that the study repeated with this change could demonstrate statistically significant menu type effect on lateral lane holding performance.

Given 3VP limitations, it may be interesting to conduct a further study with driving tasks involving a realistic simulated scenario such as navigating to a particular location while obeying traffic controls and interacting with other drivers.

This study is consistent with the previous studies in regards to menu accuracy and response rates, as well as perceived workload. Biometric measures of workload were desired as such data would likely help distinguish between the impact of user preference of menu type on perceived workload estimation. Regrettably, difficulties with measurement equipment made these comparisons impossible for the study.

In summary, this study has been in line with the previous dual task experiment (Study 2) and has demonstrated some preliminary results regarding the positive effects that the push menu type can have on driving tasks.

CHAPTER 9

DESIGN GUIDELINES

Through the efforts of this research, a number of design guidelines have been identified. Most pertinent is that push menus appear to be primarily useful in dual task scenarios. In isolated menu interaction tasks, users are frustrated by the lack of control of the presentation of the menu items. However, when a user is focused on a demanding visual-motor primary task, the frustrations are no longer as much a concern as the user is so engaged with the primary task. Ultimately push menus seem to preserve much higher MRT resources for the primary task as opposed to the resource contention caused by pull menus.

Push menus also have the advantage of requiring a minimal physical interface of just one button. This can be advantageous for devices with limited physical real estate such as a wearable device, devices that must be extremely low cost, or devices that must be extremely easy to use with universally understood affordances.

Push menus also seem most appropriate for short to mid-length menus of around fifteen items or less. Longer menus are more likely to be frustrating for users. It may be possible to break up larger lists of items in a hierarchy, but menu designers run the risk of users getting lost within the structure. Additionally, a menu designer could increase the presentation rate of a push menu but there is a risk of increasing selection error.

A push menu with auditory representation is largely not affected in user performance by the presence of a multimodal visual menu. The user cannot make a push menu go any faster and thus the user cannot take advantage of the benefits of visual scanning like can be done with a visual pull menu. Regardless, it is reasonable to consider blanking an interface screen while a user is interacting with a push menu and focused on another primary task with visual demands. For activities such as operating a vehicle, this lack of competing visual can be of great benefit and the auditory representation provided sufficient means of interaction. Further research is needed to determine if users are less inclined to look at menu visuals when engaged with push menus as compared with pull menus.

A variety of aspects of push menu auditory rendering can impact the speed of interaction. Ideally, a menu designer will want to speed up interaction without significant detriment to menu item selection accuracy. The previous studies identify silence interval duration and selection overlap as two effective attributes that can be adjusted for optimal speed/accuracy trade-off. The ideal values in the context of the controlled experiments presented are 200ms silence duration and 200ms selection overlap.

Furthermore, a menu designer may consider speeding up the playback of the actual menu items via pitch-corrected time compression of the audio stream. While not explored in this research, this may be a further useful dimension in the optimization of push menu throughput.

Menu familiarity is anticipated to confound efforts to increase the presentation speed of a push menu. For instance, menu designers might draw upon generalized characteristics of spoken word recognition in informing push menu overlap duration (see Study 4). However, one may imagine a number of possibilities in which a user is interacting with a menu. These possibilities may include whether the user knows exactly which menu item she seeks (e.g. “Air Conditioner Control”) or perhaps the user must make a decision as to which category is most relevant to an interface goal (e.g. the user wants to change the fan speed of the air conditioner with an unfamiliar vehicle interface and doesn't know the menu item title). Additionally, the user may or may not be familiar with the complete menu and options, including the order of presentation. Related to this, menus may reorder themselves based on frequency of use or other contextual information. Collectively, these variations on menu use case can impact the most appropriate optimizations of the push menu.

For instance, it is hypothesized that a user that knows exactly which menu item is desired but that is not familiar with the entirety of the menu or the order of items will react similarly to previous spoken word recognition studies (Allopenna, Magnuson, & Tanenhaus, 1998). However, if the user is further aware of the menu order then she might instead respond more quickly than spoken word recognition models predict. In this case, a menu designer may need to reduce push menu overlap duration. If a user is not aware of the exact menu item desired, she may choose to listen to all options once before making a selection of the second go-round. Again, this may result in behavior that differs from the other scenarios above.

In light of this potentially variable response, it may be desirable to dynamically model user capabilities in push menu interaction. Such a system may adaptively adjust push menu characteristics so as to optimize menu throughput. Such optimizations could be bounded by recognition of menu error rate, but this is not necessarily straightforward to detect unless there are well-defined corrective user inputs or confirmations of action that the user must submit.

Prediction of the user's push menu interaction capability may be further modeled in light of MRT. For instance, if the menu system is aware that a vehicle operator is engaged with a particularly difficult driving task then overall push menu throughput could be reduced accordingly to anticipated contention at various MRT stages of processing.

While the research presented in this thesis did not utilize an adaptive menu system, experimental results do suggest that optimal push menu configuration does vary with individual ability and context (e.g. menu familiarity and multitasking demands). Assuming a push menu designer is not able to create an adaptive system, then the push menu characteristics should be conservatively set such that the menu is generally accessible to the target user. A user configurable push menu throughput rate could be a useful compromise to satisfying varying user ability.

One other design approach worth consideration is the notion of switching between push and pull menus as appropriate. This concept might work by allowing a user to interact with a pull menu when singularly focused on the menu system interaction. However, if demands on the user change then the menu system can be switched to a push menu. This menu type switch can be initiated directly by the user, or automatically triggered by some contextual measurement (e.g. continuous measure of primary task difficulty, user workload, stress, etc.) This is a potentially powerful approach to leveraging the advantages of both menu interactions.

Menu designers may desire to make efforts to further improve push menu selection accuracy through additional inputs from the user to confirm/correct the selection. Study 3 discussed one such approach of allowing a sort of rewind function when a user overshoots. Many other methods may be effective approaches. For instance, the menu system could include a confirmation of selection (e.g. “Select *Air Conditioner*? Yes…No.”). Another option is to repeat push menu options temporally close to the user selection event. Likely, the push menu would play at a slower speed for this secondary confirmation selection. This approach is somewhat similar to the fast forward feature on digital video recorders, which when a user selects to end a fast-forward operation moves the play cursor back in the timeline to compensate for delayed user response. Another approach may be to support a jog wheel style interface where the user can pick a speed of menu playback dynamically.

Any of the above approaches to allowing the user to clarify push menu selections may be effective in improving menu selection accuracy. However, menu designers should be cautious in that any of these techniques can be abused in that a user can become equally as engaged with physical menu interactions as a pull menu. In fact, Study 3 clearly indicated an unintended emergent abuse of the menu overshoot correction feature by some participants.

There are a variety of application areas in which it may be appropriate to apply push menus. The research presented suggests vehicle operation as a particularly important area. Naturally, this applicability can be extended to all kinds of vehicles, especially those in which the operator is continuously engaged in applying input control. Aircraft such as airplanes and especially helicopters could benefit from push menus.

Additionally personal wearable devices could see benefit from push menus. Users may be variously engaged with the demands of their activities and surroundings. Devices such as smart earpieces have extremely small form factors and hardly room for a single button. In this case, a push menu may be particularly appropriate.

In summary, the push menu concept offers a broad range of advantages across a number of potential application areas. There are a number of design dimensions that can be adjusted as appropriate and some informed recommendations are presented. There are still many aspects of push menus yet to explore, particularly related to dynamically adjusting the menu parameters according to context. However, push menus can clearly be effective with the methods presented.

CHAPTER 10

CONCLUSION

This thesis traces a focused research thread exploring the interface modality of auditory display as applied to menu selection activities that must be performed simultaneously with heavy contention on perceptual, cognitive, and response processing stages of task completion. This work goes beyond simply swapping visual interface widgets with equivalent auditory constructs. Instead it considers the demands of multitasking and attempts to diminish the dissonance created by antagonistic dual tasks.

In particular, four key contributions have been made. First, an empirically grounded design space of push menus was presented. Second, a theoretical basis within the context of Multiple Resource Theory was given, explaining the benefits of push menu interaction during demanding primary tasks. Third, a flexible and functional auditory menu system was developed supporting both push and pull menus as well as established auditory display capabilities. Lastly, the effectiveness of push menus in dual task scenarios was demonstrated with statistically significant evidence supporting that hypothesis.

It is clear that there are tradeoffs between the two menu types. Push menus concede overall efficiency, perceived workload, and preference to pull menus. However, there is significant evidence that push menus can have less negative impact on a critical visual motor primary task. This has shown to be true even when users must stay engaged with a push menu for a longer period of time than a pull menu for an otherwise equivalent selection task (due to the longer selection time).

Pull menu interactions can be heavily shaped by users through preemptive navigation actions, allowing for significant process optimization. Push menus however can only be optimized through playback speed and interpretation of user response at the system level. The overall playback speed can be adjusted by the silence interval between menu items as well as the actual menu item's presentation rate (perhaps pitch corrected time compression). Leveraging the fact that it takes a user some amount of time to recognize a menu item, the selection intervals can also be optimized to improve menu selection accuracy with little or no negative impact on accuracy rates (as detailed previously in discussion of selection overlap). This effort to optimize playback rate can likely be further improved through dynamic adjustment of speed, perhaps driven by user feedback, either directly though user setting or indirectly via behavior analysis.

Push menus may also benefit from the ability to manually refine interactions. An initial effort was presented in the third study via the correction interaction. However, any such interaction runs the risk of devolving into a pull menu through abuse of the feature if not properly constrained. Another possibility not explored in these studies but presented is the concept of selection confirmation. A selection confirmation could allow a user to make a selection, and then be presented with a second opportunity for input confirming the selection. An absence of follow-up confirmation could cancel the selection. This could also reduce some of the frustrations users have identified. However, every additional input required from the user runs the risk of negative impact on the primary task.

The success of the push menu concept as compared to pull menus is supported by the Multiple Resource Theory. At minimum, there appears to be a reduction of resource contention at the response processing stage. The fact that users engaged with a pull menu must rapidly transition from auditory perceptual to cognitive decision making to tactile response over and over suggests that there may also be a context switch penalty. This is in comparison to a push menu interaction where a user is primarily engaged with auditory perceptual and cognitive processing until the target menu item is recognized and a response is required.

Push menus have proven straightforward to implement and should be easy to integrate into any event based user interface system with robust audio output capabilities. The research effort detailed in this thesis resulted in an implementation that supported both the System for Wearable Audio Navigation (SWAN) as well as the user study platform. This begins to show the flexibility of the interface components associated with push menus.

In conclusion, push menus are a viable auditory interface for use in dual task scenarios involving significant visual motor primary tasks. The most immediate application is in vehicles with peripheral computer systems such as an infotainment center console. Push menus could be equally applicable for users that have visual impairments or are temporarily unable to safely look at a visual interface.

APPENDIX A

SCREEN CAPTURES FROM EXPERIMENT SOFTWARE

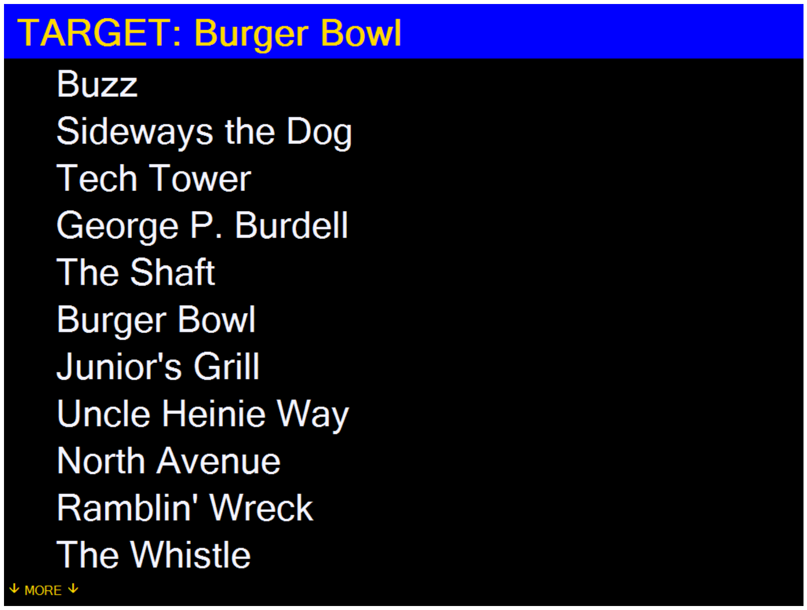

Figure A.1: A screen capture of the experiment software with a menu of items.

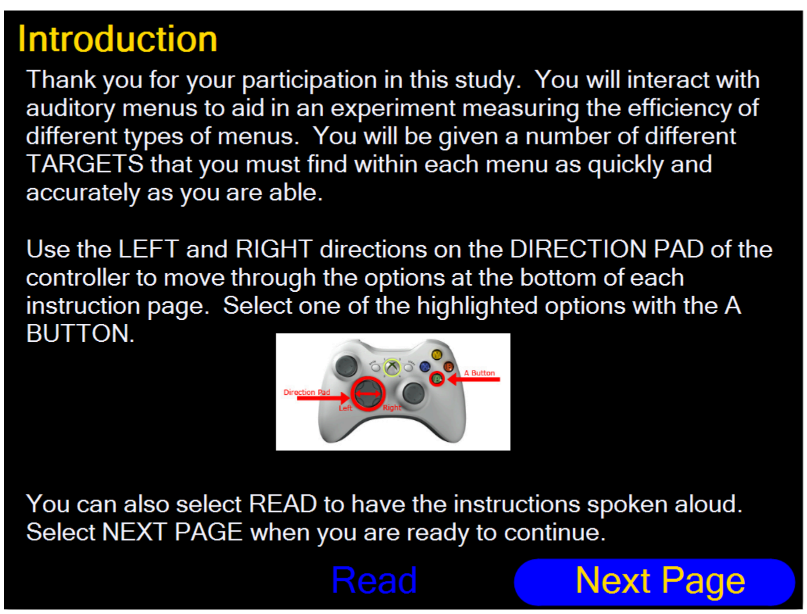

Figure A.2: The experiment software with built in multimedia instructional pages.

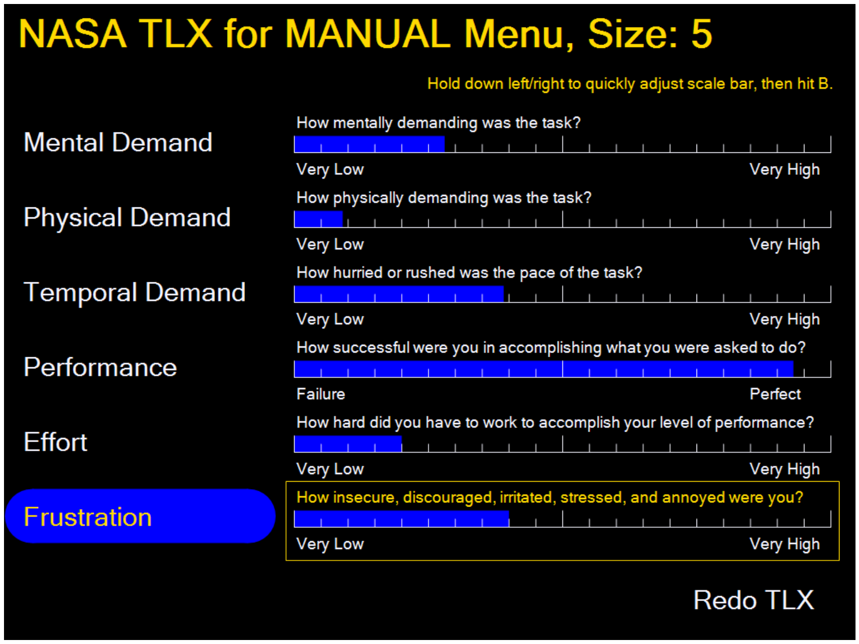

Figure A.3: Experiment software with built-in NASA TLX dimension ranking.

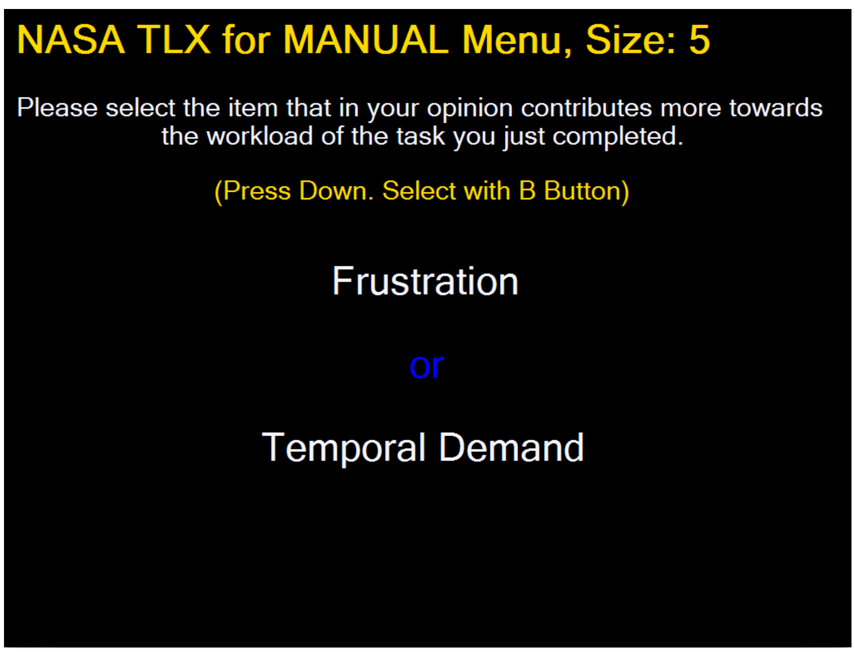

Figure A.4: Experiment software with built-in NASA TLX dimension ratings.

APPENDIX B

STUDY 1 – EXTENDED RESULTS

Table B.1: Menu Type * Menu Size t-Tests

Measure	Menu Size	Menu Type A	Menu Type B	t(28)	p (corrected)	p < 0.05 ?
Trial Duration	size 5	pull	push	-15.61	.00	1
		push	pull	15.61	.00	1
	size 20	pull	push	-22.07	.00	1
		push	pull	22.07	.00	1
	size 40	pull	push	-31.40	.00	1
		push	pull	31.40	.00	1

Table B.2: Menu Type * Menu Size t-Tests (2)

Measure	Menu Type	Menu Size A	Menu Size B	t(28)	p (corrected)	p < 0.05 ?
duration	pull	size 5	size 20	-13.36	.00	1
		size 5	size 40	-21.97	.00	1
		size 20	size 40	-19.17	.00	1
	push	size 5	size 20	-45.21	.00	1
		size 5	size 40	-53.68	.00	1
		size 20	size 40	-44.01	.00	1

Table B.3: Menu Type * Visual Condition Trial Duration Descriptive Statistics

Menu Type	Visual Condition	Mean	Std. Dev.
pull	off	9.75	1.98
	on	4.21	1.85
push	off	16.81	1.83
	on	15.97	1.70

Table B.4: Menu Size * Visual Condition Trial Duration Descriptive Statistics

Menu Size	Visual Condition	Mean	Std. Dev.
size 5	off	3.41	0.44
	on	2.57	0.41
size 20	off	9.69	1.38
	on	7.78	1.28
size 40	off	26.74	3.42
	on	19.93	3.19

Table B.5: Menu Type * Visual Condition on Selection Duration t-Tests

Visual Condition	Menu Type	Menu Type	t(28)	p (corrected)	p < 0.05 ?
off	1	2	-17.40	.00	1
on	1	2	-31.11	.00	1

Table B.6: Menu Type * Visual Condition on Trial Duration t-Tests (2)

Menu Type	Visual Condition A	Visual Condition B	t(28)	p (corrected)	p < 0.05 ?
pull	off	on	10.81	.00	1
push	off	on	1.79	.34	0

Table B.7: Menu Size * Visual Condition on Trial Duration t-Tests (3)

Visual Condition	Menu Size A	Menu Size B	t(28)	p (corrected)	p < 0.05 ?
off	size 5	size 20	-27.31	.00	1
	size 5	size 40	-37.57	.00	1
	size 20	size 40	-34.52	.00	1
on	size 5	size 20	-24.34	.00	1
	size 5	size 40	-30.04	.00	1
	size 20	size 40	-26.42	.00	1

Table B.8: Menu Size * Visual Condition on Trial Duration T-Tests (4)

Menu Size	Visual Condition A	Visual Condition B	t(28)	p (corrected)	p < 0.05 ?
size 5	off	on	7.32	.00	1
size 20	off	on	5.38	.00	1
size 40	off	on	7.71	.00	1

Table B.9: Menu Type * Menu Size * Visual Cond. Trial Duration Descriptive Statistics

Menu Type	Menu Size	Visual Condition	Mean	Std. Dev.
pull	size 5	off	2.86	0.61
		on	1.60	0.57
	size 20	off	7.09	2.05
		on	3.63	1.91
	size 40	off	19.29	4.11
		on	7.41	3.83
push	size 5	off	3.95	0.55
		on	3.54	0.51
	size 20	off	12.29	1.55
		on	11.93	1.44
	size 40	off	34.19	4.33
		on	32.45	4.03

Table B.10: Menu Type * Menu Size * Visual Condition on Trial Duration t-Tests

Menu Size	Visual Condition	Menu Type A	Menu Type B	t(28)	p (corrected)	p < .05 ?
1	0	1	2	-7.68	.00	1
	1	1	2	-14.69	.00	1
2	0	1	2	-11.62	.00	1
	1	1	2	-19.90	.00	1
3	0	1	2	-16.00	.00	1
	1	1	2	-28.88	.00	1

Table B.11: Menu Type * Menu Size * Visual Cond. on Trial Duration t-Tests (2)

Menu Type	Visual Condition	Menu Size A	Menu Size B	t(28)	p (corrected)	p < .05 ?
pull	off	size 5	size 20	-12.35	.00	1
		size 5	size 40	-22.20	.00	1
		size 20	size 40	-20.01	.00	1
	on	size 5	size 20	-6.38	.00	1
		size 5	size 40	-8.43	.00	1
		size 20	size 40	-6.66	.00	1
push	off	size 5	size 20	-30.89	.00	1
		size 5	size 40	-37.47	.00	1
		size 20	size 40	-31.06	.00	1
	on	size 5	size 20	-33.42	.00	1
		size 5	size 40	-38.50	.00	1
		size 20	size 40	-31.29	.00	1

Table B.12 – Menu Type * Menu Size * Visual Cond. on Trial Duration t-Tests (3)

Menu Type	Menu Size	Visual Condition A	Visual Condition B	t(28)	p (corrected)	p < .05 ?
pull	size 5	off	on	8.06	.00	1
	size 20	off	on	6.53	.00	1
	size 40	off	on	11.20	.00	1
push	size 5	off	on	2.92	.17	0
	size 20	off	on	0.92	1.0	0
	size 40	off	on	1.56	1.0	0

APPENDIX C

STUDY 2 – EXTENDED RESULTS

Table C.1: Menu Type * Menu Size Descriptive Statistics

Measure	menuType	menuSize	Mean	Std. Dev
gameAccuracy	**pull**	5	.88	.08
		20	.83	.10
		40	.80	.12
	push	5	.90	.08
		20	.87	.11
		40	.86	.10
trialDuration	**pull**	5	3.49	0.97
		20	8.76	2.01
		40	20.06	6.55
	push	5	3.99	0.57
		20	12.49	1.75
		40	33.55	3.11
menuAccuracy	**pull**	5	.95	.08
		20	.95	.09
		40	.95	.05
	push	5	.93	.05
		20	.92	.07
		40	.86	.15

Table C.2: Menu Type * Menu Size t-tests

Measure	menuType	menuSize A	menuSize B	$t(35)$ =	p (corrected)	$p < .05$?
gameAcc	**pull**	5	20	7.00	.00	1
		5	40	10.25	.00	1
		20	40	4.13	.00	1
	push	5	20	3.00	.03	1
		5	40	5.57	.00	1
		20	40	1.88	.76	0
trialDuration	**pull**	5	20	-17.45	.00	1
		5	40	-15.54	.00	1
		20	40	-11.44	.00	1
	push	5	20	-31.36	.00	1
		5	40	-56.41	.00	1
		20	40	-45.00	.00	1
menuAccuracy	**pull**	5	20	0.27	1.00	0
		5	40	0.08	1.00	0
		20	40	-0.23	1.00	0
	push	5	20	1.33	1.00	0
		5	40	3.27	.03	1
		20	40	2.43	0.20	0

Table C.3: Menu Type * Menu Size t-tests (2)

Measure	menuSize	menuType A	menuType B	*t*(35)	*p* (corrected)	*p* < .05 ?
gameAcc	5	pull	push	-1.75	.76	0
	20			-3.90	.01	1
	40			-5.18	.00	1
trialDur	5	pull	push	-3.05	.05	1
	20			-11.21	.00	1
	40			-11.66	.00	1
menuAcc	5	pull	push	1.47	1.00	0
	20			2.20	.28	0
	40			4.04	.00	1

APPENDIX D

STUDY 3 – EXTENDED RESULTS

Table D.1: Menu Size * Silence Duration Interaction Descriptive Statistics

Measure	Menu Size	Silence Duration	Mean	Std. Dev.
Accuracy	size 5	200ms	.92	.09
		400ms	.96	.04
		600ms	.98	.03
	size 20	200ms	.86	.07
		400ms	.96	.04
		600ms	.97	.03
Duration	size 5	200ms	3.22	0.31
		400ms	3.63	0.20
		600ms	4.07	0.20
	size 20	200ms	9.59	0.53
		400ms	11.39	0.54
		600ms	13.36	0.53

Table D.2: Menu Size * Silence Duration Interaction t-tests

Measure	Menu Size	Silence Duration A	Silence Duration	*t*(22)	*p*(corrected)	*p* < .05 ?
accuracy	size5	200ms	400ms	-2.00	.52	0
		200ms	600ms	-2.95	.07	0
		400ms	600ms	-1.55	1.00	0
	size20	200ms	400ms	-7.23	0.00	1
		200ms	600ms	-7.00	0.00	1
		400ms	600ms	-0.91	1.00	0
duration	size5	200ms	400ms	-7.83	0.00	1
		200ms	600ms	-13.36	0.00	1
		400ms	600ms	-11.89	0.00	1
	size20	200ms	400ms	-18.97	0.00	1
		200ms	600ms	-33.63	0.00	1
		400ms	600ms	-16.37	0.00	1

Table D.3: Menu Size * Silence Duration Interaction t-tests (2)

Measure	Silence Duration	Menu Size A	Menu Size B	*t*(22)	*p*(corrected)	*p* < .05 ?
accuracy	200ms	size5	size20	4.46	0.00	1
	400ms	size5	size20	0.55	1.00	0
	600ms	size5	size20	1.44	1.00	0
duration	200ms	size5	size20	-73.25	0.00	1
	400ms	size5	size20	-76.83	0.00	1
	600ms	size5	size20	-85.17	0.00	1

APPENDIX E

STUDY 4 – EXTENDED RESULTS

Table E.1: Paired-Sample t-tests of Effects of Selection Overlap

Measure	Overlap	Overlap	t(26)	p (corrected)	p < 0.05 ?
duration	300 ms	200 ms	-10.44	0.00	1
	300 ms	0 ms	-17.73	0.00	1
	200 ms	0 ms	-12.10	0.00	1
workload	300 ms	200 ms	1.74	0.28	0
	300 ms	0 ms	3.08	0.02	1
	200 ms	0 ms	1.82	0.24	0

Table E.2: Menu Size * Selection Overlap Descriptive Statistics

Measure	Menu Size	Overlap	Mean	Std. Dev.
Duration	size 5	300 ms	2.98	0.28
		200 ms	3.22	0.34
		0 ms	3.62	0.28
	size 20	300 ms	8.61	0.52
		200 ms	9.52	0.48
		0 ms	11.58	0.78

Table E.3: Menu Size * Selection Overlap t-tests

Overlap	Menu Size A	Menu Size B	t(26)	p (corrected)	p < 0.05 ?
300 ms	size 5	size 20	-51.61	0.00	1
200 ms	size 5	size 20	-67.01	0.00	1
0 ms	size 5	size 20	-65.81	0.00	1

Table E.4: Menu Size * Selection Overlap t-tests (2)

menuSize	Overlap A	Overlap B	t(26)	p (corrected)	p < 0.05 ?
size 5	300 ms	200 ms	-3.39	0.02	1
	300 ms	0 ms	-10.83	0.00	1
	200 ms	0 ms	-5.09	0.00	1
size 20	300 ms	200 ms	-8.13	0.00	1
	300 ms	0 ms	-16.63	0.00	1
	200 ms	0 ms	-13.33	0.00	1

REFERENCES

Allopenna, P. D., Magnuson, J. S., & Tanenhaus, M. K. (1998). Tracking the Time Course of Spoken Word Recognition Using Eye Movements: Evidence for Continuous Mapping Models. *Journal of Memory and Language*, *38*(4), 419–439. https://doi.org/10.1006/jmla.1997.2558

Arons, B. (1997). SpeechSkimmer: a system for interactively skimming recorded speech. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *4*(1), 3–38.

Audacity: Free Audio Editor and Recorder. (n.d.). Retrieved April 5, 2014, from http://audacity.sourceforge.net/

Baldwin, C. L. (2011). Verbal collision avoidance messages during simulated driving: perceived urgency, alerting effectiveness and annoyance. *Ergonomics*, *54*(4), 328–337. https://doi.org/10.1080/00140139.2011.558634

Boren, R. W., Moor, W. C., & Anderson-Rowland, M. R. (1997). Optimal number of choices per menu for naive users of a computer-answered telephone system. *Computers & Industrial Engineering*, *32*(2), 363–370.

Brewster, S. A., Wright, P. C., & Edwards, A. D. N. (1993). An evaluation of earcons for use in auditory human-computer interfaces. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (pp. 222–227). New York, NY, USA: ACM. https://doi.org/10.1145/169059.169179

Cepstral - Text to Speech for Personal, Business, and Enterprise Use. (n.d.). Retrieved April 5, 2014, from http://www.cepstral.com/

Cherry, E. C. (1953). Some Experiments on the Recognition of Speech, with One and with Two Ears. *The Journal of the Acoustical Society of America*, *25*(5), 975–979. https://doi.org/10.1121/1.1907229

Cohen, M., & Ludwig, L. F. (1991). Multidimensional audio window management. *International Journal of Man-Machine Studies*, *34*(3), 319–336. https://doi.org/10.1016/0020-7373(91)90023-Z

Colven, D., & Judge, S. (1996). *Switch Access to Technology a Comprehensive Guide*. ACE Centre Oxford.

Dowburd, C. (1994). Touchtone Telephone Registration Through A Service Bureau Network. In *Proceedings of the 5th Annual Connectivity and Technology Symposium*. West Chester, PA: DIANE Publishing.

Edwards, W. K., & Mynatt, E. D. (1994). An architecture for transforming graphical interfaces. In *Proceedings of the 7th annual ACM symposium on User interface software and technology* (pp. 39–47). Marina del Rey, California, United States: ACM. Retrieved from http://portal.acm.org.www.library.gatech.edu:2048/citation.cfm?id=192426.192443&coll=portal&dl=ACM&CFID=21710409&CFTOKEN=58786272

Gaver, W. W. (1986). Auditory Icons: Using Sound in Computer Interfaces. *Human-Computer Interaction*, *2*, 167–177.

Helal, A., Moore, S. E., & Ramachandran, B. (2001). Drishti: an integrated navigation system for visually impaired and disabled. In *Fifth International Symposium on Wearable Computers, 2001. Proceedings* (pp. 149–156). https://doi.org/10.1109/ISWC.2001.962119

Henja, D., & Musicus, B. (1991). The solafs time-scale modification algorithm. *Bolt, Beranek and Newman (BBN) Technical Report*.

Hermann, T., & Ritter, H. (1999). Listen to your data: Model-based sonification for data analysis. *Advances in Intelligent Computing and Multimedia Systems*, *8*, 189–194.

Horberry, T., Anderson, J., Regan, M. A., Triggs, T. J., & Brown, J. (2006). Driver distraction: The effects of concurrent in-vehicle tasks, road environment complexity and age on driving performance. *Accident Analysis & Prevention*, *38*(1), 185–191. https://doi.org/10.1016/j.aap.2005.09.007

IBM The calculation of Bonferroni-adjusted p-values - United States. (2012, April 20). [CT741]. Retrieved April 20, 2014, from http://www-01.ibm.com/support/docview.wss?uid=swg21476685

Jeon, M., & Walker, B. N. (2009). “Spindex”: Accelerated Initial Speech Sounds Improve Navigation Performance in Auditory Menus. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *53*(17), 1081–1085. https://doi.org/10.1177/154193120905301708

Kahneman, D. (1973). *Attention and effort*. Englewood Cliffs, NJ: Prentice Hall.

Klapp, S. T., & Nelson, J. M. (1998). Can People Tap Concurrent Bimanual Rhythms Independently? *Journal of Motor Behavior*, *30*(4), 301.

Klatzky, R. L., Marston, J. R., Giudice, N. A., Golledge, R. G., & Loomis, J. M. (2006). Cognitive load of navigating without vision when guided by virtual sound versus spatial language. *Journal of Experimental Psychology: Applied*, *12*(4), 223–232. https://doi.org/10.1037/1076-898X.12.4.223

Loomis, J. M., Golledge, R. D., & Klatzky, R. L. (2001). GPS-based navigation systems for the visually impaired. In W. Barfield & T. Caudell (Eds.), *Fundamentals of wearable computers and augmented reality* (pp. 429–446). Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers.

Loomis, J. M., Klatzky, R. L., & Golledge, R. G. (2001). Navigating without vision: basic and applied research. *Optometry & Vision Science*, *78*(5), 282–289.

MacIntyre, B., & Feiner, S. (1996). Future multimedia user interfaces. *Multimedia Systems*, *4*(5), 250–268. https://doi.org/10.1007/s005300050027

Marila, J. (2002). Experimental comparison of complex and simple sounds in menu and hierarchy sonification. In *Proceedings of the International Conference on Auditory Display* (pp. 104–108).

Martens, M., & Van Winsum, W. (2000). Measuring distraction: the peripheral detection task. *TNO Human Factors, Soesterberg, Netherlands*.

Miller, D. P. (1981). The depth/breadth trade-off in hierarchical computer menus. *Proceedings of the Human Factors Society - 25th Annual Meeting*, 296–300.

Mowbray, G. H. (1953). Simultaneous vision and audition: the comprehension of prose passages with varying levels of difficulty. *Journal of Experimental Psychology*, *46*(5), 365.

Mynatt, E. D. (1992). Auditory presentation of graphical user interfaces. In *Proceedings of the 1st International Conference on Auditory Display* (pp. 533–555). Atlanta, Ga.

National Highway Traffic Safety Administration. (2012a). Blueprint for ending distracted driving. *Washington, DC: US Department of Transportation. National Highway Traffic Safety Administration*.

National Highway Traffic Safety Administration. (2012b). *Visual-manual NHTSA driver distraction guidelines for in-vehicle electronic devices* (No. NHTSA-2010-0053). Washington, DC: National Highway Traffic Safety Administration (NHTSA), Department of Transportation (DOT).

Nees, M. A., & Walker, B. N. (2011). Auditory Displays for In-Vehicle Technologies. *Reviews of Human Factors and Ergonomics*, *7*(1), 58–99. https://doi.org/10.1177/1557234X11410396

Palladino, D. K. (2008, May). Efficiency of Spearcon-Enhanced Navigation of One Dimensional Electronic Menus [Undergraduate Thesis]. Retrieved November 11, 2014, from https://smartech.gatech.edu/handle/1853/26302

Palladino, D. K., & Walker, B. N. (2007). Learning Rates for Auditory Menus Enhanced with Spearcons versus Earcons. *Proceedings of the 13th International Conference on Auditory Display*, (Montreal, Canada, June 26-29, 2007), 274–279.

Ran, L., Helal, S., & Moore, S. (2004). Drishti: an integrated indoor/outdoor blind navigation system and service. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications, 2004. PerCom 2004* (pp. 23–30). https://doi.org/10.1109/PERCOM.2004.1276842

Resnick, P., & Virzi, R. A. (1992). Skip and scan: cleaning up telephone interface (pp. 419–426). Presented at the Proceedings of the SIGCHI conference on Human factors in computing systems, ACM.

Resnick, P., & Virzi, R. A. (1995). Relief from the Audio Interface Blues: Expanding the Spectrum of Menu, List, and Form Styles. *ACM Transactions on Computer-Human Interaction*, *2*(2), 145–176.

Sawhney, N., & Schmandt, C. (2000). Nomadic radio: speech and audio interaction for contextual messaging in nomadic environments. *ACM Trans. Comput.-Hum. Interact.*, *7*(3), 353–383.

Schmandt, C. (1993). *Voice Communication with Computers: Conversational Systems*. New York, NY: Van Nostrand Reinhold.

Starner, T. (2001). The challenges of wearable computing: Part 1. *IEEE Micro*, *21*(4), 44–52. https://doi.org/10.1109/40.946681

Stifelman, L. J., Arons, B., Schmandt, C., & Hulteen, E. A. (1993). VoiceNotes: a speech interface for a hand-held voice notetaker. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems* (pp. 179–186). ACM New York, NY, USA.

Strothotte, T., Fritz, S., Michel, R., Raab, A., Petrie, H., Johnson, V., … Schalt, A. (1996). Development of dialogue systems for a mobility aid for blind people: initial design and usability testing. In *Proceedings of the second annual ACM conference on Assistive technologies* (pp. 139–144). New York, NY, USA: ACM. https://doi.org/10.1145/228347.228369

Walker, B. N. (2007). Consistency of magnitude estimations with conceptual data dimensions used for sonification. *Applied Cognitive Psychology*, *21*(5), 579–599.

Walker, B. N., & Cothran, J. T. (2003). Sonification Sandbox: A graphical toolkit for auditory graphs. In *Proceedings of the Ninth International Conference on Auditory Display* (pp. 161–163). Boston, MA.

Walker, B. N., & Lindsay, J. (2005). The effect of a speech discrimination task on navigation in a virtual environment. In *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting* (pp. 1538–1541).

Walker, B. N., Nance, A., & Lindsay, J. (2006). Spearcons: Speech-based Earcons Improve Navigation Performance in Auditory Menus. *Proceedings of the International Conference on Auditory Display (ICAD 2006)*, 63–68.

Wenzel, E. M., Arruda, M., Kistler, D. J., & Wightman, F. L. (1993). Localization using nonindividualized head-related transfer functions. *The Journal of the Acoustical Society of America*, *94*(1), 111–123. https://doi.org/10.1121/1.407089

Wickens, C. D. (2000). *Engineering psychology and human performance* (3rd ed). Upper Saddle River, NJ: Prentice Hall.

Wickens, C. D. (2002). Multiple resources and performance prediction. *Theoretical Issues in Ergonomics Science*, *3*(2), 159–177. https://doi.org/10.1080/14639220210123806

Wickens, C. D., Dixon, S. R., & Seppelt, B. (2005). Auditory preemption versus multiple resources: Who wins in interruption management? (Vol. 49, pp. 463–466). Presented at the Proceedings of the Human Factors and Ergonomics Society Annual Meeting, SAGE Publications.

Wilson, J., Walker, B. N., Lindsay, J., Cambias, C., & Dellaert, F. (2007). SWAN: System for Wearable Audio Navigation. In *2007 11th IEEE International Symposium on Wearable Computers* (pp. 91–98). https://doi.org/10.1109/ISWC.2007.4373786

Yalla, P., & Walker, B. N. (2007). Advanced Auditory Menus [Technical Report]. Retrieved February 16, 2009, from http://smartech.gatech.edu/handle/1853/26170

VITA

Jeff Wilson

Jeff Wilson was born in the state of Georgia. He attended public schools in Buford, Georgia, received a B.S. in Computer Science from Georgia Tech in 1999, and a M.S. in Computer Science from Georgia Tech in 2001. Mr. Wilson was employed as a research scientist in the Biomedical Interactive Technology Center of Georgia Tech in 2001 and later transitioned to working at the Interactive Media Technology Center of Georgia Tech where he is currently employed as a senior research scientist. Mr. Wilson's hobbies include powerlifting, bicycling, video games, and spending time with his wife and children.